The Knowledge Gradient for Sequential Decision Making with Stochastic Binary Feedbacks

Yingfei Wang YINGFEI@CS.PRINCETON.EDU
Department of Computer Science, Princeton University, Princeton, NJ 08540

Chu Wang CHUW@MATH.PRINCETON.EDU
The Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544

Warren Powell POWELL@PRINCETON.EDU
Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544

Abstract

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. Our problem is motivated by real-world applications where observations are time consuming and/or expensive. With the adaptation of an online Bayesian linear classifier, we develop a knowledge-gradient type 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 provide a finite-time analysis of the estimated error and demonstrate the performance of the proposed algorithm on both synthetic problems and benchmark UCI datasets.

1. Introduction

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.

The problem of deciding which medical decisions to evaluate can be modeled mathematically as a sequential decision making problem with stochastic binary outcomes. In this setting, we have a small budget of measurements that we allocate sequentially to medical decisions so that after the budget exhausted, we have collected information to maximize our ability to choose the medical decision with the highest probability of success.

Scientists can draw on an extensive body of literature on the classic design of experiments (DeGroot, 1970; Wetherill & Glazebrook, 1986; Montgomery, 2008) whose goal is to decide what observations to make when fitting a function. Yet in our settings, the decisions are guided by a well-defined utility function (that is, maximize the probability of success). This problem also relates to active learning (Schein & Ungar, 2007; Tong & Koller, 2002; Freund et al., 1997; Settles, 2010). Our model is most similar to membership query synthesis where the learner may request labels for unlabeled instances 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. Other relevant and yet different works include budgeted learning to imitate the oracle's behavior (He et al., 2012), and adaptive selection of

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

pre-trained classifiers (Gao & Koller, 2011).

Another similar setting is multi-armed bandit problems (Auer et al., 2002; Bubeck & Cesa-Bianchi, 2012; Filippi et al., 2010; Mahajan et al., 2012; Chapelle & Li, 2011; Li et al., 2010) for cumulative regret minimization in an online setting. Our work will initially focus on offline settings such as laboratory experiments or medical trials where we are not punished for errors incurred during training and only concern with the final recommendation after the offline training phases. The knowledge gradient for offline learning extends easily to bandit settings with the goal to minimize the cumulative regret (Ryzhov et al., 2012). There are works to address the problem we describe here by minimizing the simple regret. But first, the UCB type policies (Audibert et al., 2010) are not best suited for expensive experiments. Second, the work on simple regret minimization (Hoffman et al., 2014; Hennig & Schuler, 2012) mainly focuses on real-valued functions.

There is a literature on Bayesian optimization (He et al., 2007; Chick, 2001; Powell & 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 & 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.

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 variant designed for binary classification with parametric belief models. A particularly relevant work in the Bayesian optimization literature is the expected improvement (EI) for binary outputs (Tesch et al., 2013). EI is an approximation of KG assuming no measurement noise (see Section 5.6 of (Powell & Ryzhov, 2012) and (Huang et al., 2006) for detailed explanations). In our setting with stochastic binary outcomes, the stochasticity is not explicitly considered by the EI calculation.

The main contributions of this paper are organized as follows. We first rigorously establish a sound mathematical model for the problem of sequentially maximizing the response under binary outcomes in Section 2. Due to the sequential nature of the problem, we develop an online Bayesian linear classification procedure for general link functions to recursively predict the response of each alternative in Section 4. In Section 5, we design a knowledge-gradient type policy for stochastic binary responses to guide the experiment and provide a finite-time analysis on the estimated error. This is different from the PAC (passive) learning bound which relies on the i.i.d. assumption of the examples. Extensive demonstrations and comparisons of methods are demonstrated in Section 6.

2. Problem Formulation

Given a finite set of alternatives $\boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$, where each $\boldsymbol{x}$ is represented by a d-dimensional feature vector, the observation of measuring each $\boldsymbol{x}$ is a binary outcome $y \in \{-1, +1\}/\{\text{failure, success}\}$ with some unknown probability of success $\Pr(y = +1|\boldsymbol{x})$. Under a limited budget N, our goal is to choose the measurement policy $(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N)$ and implementation decision $\boldsymbol{x}^{N+1}$ that maximizes the probability of success $\Pr(y = +1|\boldsymbol{x}^{N+1})$.

We adopt probabilistic modeling for the unknown probability of success. Under general assumptions, the posterior probability of class $+1$ can be written as a link function acting on a linear function of the feature vector

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

with the link function $\sigma(a)$ often chosen as the logistic function $\sigma(a) = \frac{1}{1+\exp(-a)}$ or probit function $\sigma(a) = \Phi(a) = \int_{-\infty}^{a} \mathcal{N}(s|0, 1^2)\mathrm{d}s$.

Adapting the concept of Gaussian processes, 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 stochastic binary outcome y^n assuming labels are generated independently given $\boldsymbol{w}$. Each alternative can be measured more than once with potentially different outcomes. Let $\mathcal{D}^n = \{(\boldsymbol{x}^i, y^i)\}_{i=0}^n$ 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 $\boldsymbol{x}^1, y^1, \ldots, \boldsymbol{x}^n, y^n$. We use $\mathcal{F}^n$ and $\mathcal{D}^n$ interchangeably. Measurement and implementation decisions $\boldsymbol{x}^{n+1}$ are restricted to be $\mathcal{F}^n$-measurable so that decisions may only depend on measurements made in the past. We use Bayes' theorem to form a sequence of posterior predictive distributions $\Pr(\boldsymbol{w}|\mathcal{D}^n)$ for $\boldsymbol{w}$ from the prior and the previous measurements.

The next lemma states the equivalence of using true probabilities and sample estimates when evaluating a policy, where Π is the set of policies. The proof is left in the supplementary material.

Lemma 1. *Let $\pi \in \Pi$ be a policy, and $\boldsymbol{x}^\pi = \arg\max_{\boldsymbol{x}} Pr(y = +1|\boldsymbol{x}, \mathcal{D}^N)$ be the implementation decision after the budget N is exhausted. Then*

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

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 1, we have (Powell & Ryzhov, 2012)

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

In other words, the optimal decision at time N is to go with our final set of beliefs. By the equivalence as stated in Lemma 1, while we want to learn the unknown true value $\max_{\boldsymbol{x}} \Pr(y=+1|\boldsymbol{x})$, we may write our problem's objective as

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

3. Background: Bayesian Linear Classfication

Linear classifiers are widely used in machine learning for binary classification (Hosmer Jr & Lemeshow, 2004). Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i$ a d-dimensional vector and $y_i \in \{-1,+1\}$, with the assumption that training labels are generated independently given $\boldsymbol{w}$, the likelihood $\Pr(\mathcal{D}|\boldsymbol{w})$ is defined as $\Pr(\mathcal{D}|\boldsymbol{w}) = \prod_{i=1}^n \sigma(y_i \cdot \boldsymbol{w}^T\boldsymbol{x}_i)$. The weight vector $\boldsymbol{w}$ is found by maximizing the likelihood of the training data $\Pr(\mathcal{D}|\boldsymbol{w})$ or equivalently, minimizing the negative log likelihood:

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

It is well-known that regularization is required to avoid over-fitting. Under l_2 regularization, the estimate of the weight vector $\boldsymbol{w}$ given by:

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

It can be shown that the log-likelihood function is globally concave in $\boldsymbol{w}$. Numerous optimization techniques are available for solving it such as steepest ascent, Newton's method and conjugate gradient ascent.

In this paper, we illustrate the ideas using the logistic link function given its analytic simplicity, but any monotonically increasing function $\sigma : \mathbb{R} \mapsto [0,1]$ can be used.

3.1. Bayesian Setup

A Bayesian approach to linear classification models requires first a prior distribution $p(\boldsymbol{w})$ for the weight parameters $\boldsymbol{w}$, from which we apply Bayes' theorem to derive the posterior $p(\boldsymbol{w}|\mathcal{D}) \propto \Pr(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$. 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 classifier is intractable. Different approximation methods can be used. In what follows, we consider the Laplace approximation. Laplace's method uses a Gaussian approximation to the posterior. It can be obtained by finding the mode of the posterior distribution and then fitting a Gaussian distribution centered at that mode (see Chapter 4.5 of (Bishop et al., 2006)). Specifically, define the logarithm of the unnormalized posterior distribution

$$\Psi(\boldsymbol{w}|\boldsymbol{m},\boldsymbol{\Sigma},\mathcal{D}) = \log\Pr(\mathcal{D}|\boldsymbol{w}) + \log\Pr(\boldsymbol{w}). \tag{3}$$

The Laplace approximation is based on a second-order Taylor expansion to Ψ around its MAP (maximum a posteriori) solution $\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}|\boldsymbol{m},\boldsymbol{\Sigma},\mathcal{D})$:

$$\Psi(\boldsymbol{w}) \approx \Psi(\hat{\boldsymbol{w}}) - \frac{1}{2}(\boldsymbol{w}-\hat{\boldsymbol{w}})^T\boldsymbol{H}(\boldsymbol{w}-\hat{\boldsymbol{w}}), \tag{4}$$

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

$$\boldsymbol{H} = -\nabla^2\Psi(\boldsymbol{w})|_{\boldsymbol{w}=\hat{\boldsymbol{w}}}. \tag{5}$$

The Laplace approximation results in a normal approximation to the posterior

$$p(\boldsymbol{w}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{w}|\hat{\boldsymbol{w}},\boldsymbol{H}^{-1}). \tag{6}$$

4. Fast Online Bayesian Linear Classification

Starting from a Gaussian prior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^0,\boldsymbol{\Sigma}^0)$, after the first n observed data, the Laplace approximated posterior distribution is $\Pr(\boldsymbol{w}|\mathcal{D}^n) \approx \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^n,\boldsymbol{\Sigma}^n)$ according to (6). 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) 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 leverage for recursive updates with each observation.

Here, we propose a fast and stable online updating formulation 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. In this recursive way of model updating, the Laplace approximated posterior is $\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^n,\boldsymbol{\Sigma}^n)$ serves as a prior to update the model when the next observation is made. By setting the batch size $n=1$ in Eq. (3) and (5), we have the sequential Bayesian linear model for classification as in Algorithm 1, 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). It is straightforward to check

Algorithm 1 Online Bayesian Linear Classification

Input: Regularization parameter $\lambda > 0$
$m_j = 0$, $q_j = \lambda$. (Each weight w_j has an independent prior $\mathcal{N}(m_j, q_j^{-1})$).
for $t = 1$ to T **do**
  Get a new point $(\boldsymbol{x}, y)$.
  Find $\hat{\boldsymbol{w}}$ as the maximizer of (3):
    $-\frac{1}{2}\sum_{j=1}^{d} q_i(w_i - m_i)^2 + \log(\sigma(y_i \boldsymbol{w}^T \boldsymbol{x}_i))$.
  $m_j = \hat{w}_j$.
  Update q_i according to (5): $q_j \leftarrow q_j - \hat{t} x_j^2$.
end for

that the sigmoid functions that are commonly used for classification problems, including logistic function, probit function, complementary log-log function and log-log function all satisfy this assumption.

We can tap a wide range of convex optimization algorithms including gradient search, conjugate gradient, and BFGS method (see (Wright & Nocedal, 1999) for details). But if we set $n = 1$ and $\Sigma = \lambda^{-1}\boldsymbol{I}$ in Eq. (3), a stable and efficient algorithm for solving

$$\arg\max_{\boldsymbol{w}} -\frac{1}{2}\sum_{j=1}^{d} q_i(w_i - m_i)^2 + \log(\sigma(y\boldsymbol{w}^T\boldsymbol{x})) \quad (7)$$

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

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

Defining p as $p := \frac{\sigma'(y\boldsymbol{w}^T x)}{\sigma(y\boldsymbol{w}^T x)}$, we have $w_i = m_i + ypx_i/q_i$. Plugging this back to 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 (7).

5. Knowledge Gradient Policy for Bayesian Linear Classification Belief Model

We are going to build on this framework to compute the knowledge gradient for a ranking and selection problem which each choice (say, a medical decision) influences the success or failure of a medical outcome.

We begin by developing the general framework for the knowledge gradient (KG) for ranking and selection problems, where the performance of each alternative is represented by a (non-parametric) lookup table model of Gaussian distribution with unknown mean and known variance. 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. By imposing a Gaussian prior $\mathcal{N}(\boldsymbol{\mu}^0, \boldsymbol{\Sigma}^0)$ on mean values of the alternatives, the posterior after the first n observations is $\mathcal{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)$. The value of a state is defined as $\max_x \mu_x^n$. 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). It enjoys nice properties, including myopic and asymptotic optimality. KG has been extended to various belief models (e.g. (Mes et al., 2011; Negoescu et al., 2011; Ryzhov et al., 2012; Wang et al., 2015)). The knowledge gradient can always be extended to online problems where we need to maximize cumulative rewards (Ryzhov et al., 2012).

Yet there is no KG variant designed for binary classification with parametric models, primarily because of the complexity 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.

5.1. Markov Decision Process Formulation

The state space $\mathcal{S}$ is the space of all possible predictive distributions $q(\boldsymbol{w})$ for $\boldsymbol{w}$. The transition function T: $\mathcal{S} \times \mathcal{X} \times \{-1, 1\}$ is defined as:

$$T\Big(q(\boldsymbol{w}), \boldsymbol{x}, y\Big) \propto q(\boldsymbol{w})\sigma(y\boldsymbol{w}^T\boldsymbol{x}). \quad (8)$$

The transition function for updating the belief state depends on the belief model $\sigma(\cdot)$ and the approximation strategy. For example, for the online Bayesian linear classifier with logistic function, the transition function can be defined as follows with a degenerate state space $\mathcal{S} := \mathbb{R}^d \times (0, \infty]^d$:

Definition 1. *The transition function T: $\mathcal{S} \times \mathcal{X} \times \{-1, 1\}$ is defined as*

$$T\Big((\boldsymbol{m}, \boldsymbol{q}), \boldsymbol{x}, y\Big) = \Big(\Big(\hat{\boldsymbol{w}}(\boldsymbol{m}), \boldsymbol{q} + p(1-p)\boldsymbol{diag}(\boldsymbol{x}\boldsymbol{x}^T)\Big)\Big),$$

where $\hat{\boldsymbol{w}}(\boldsymbol{m}) = \arg\min_{\boldsymbol{w}} \Psi(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{q}, (\boldsymbol{x}, y))$, $p = \sigma(\hat{\boldsymbol{w}}^T\boldsymbol{x})$ and $\boldsymbol{diag}(\boldsymbol{x}\boldsymbol{x}^T)$ is a column vector containing the diagonal elements of $\boldsymbol{x}\boldsymbol{x}^T$, so that $S^{n+1} = T(S^n, \boldsymbol{x}, Y^{n+1})$.

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. In the case of stochastic binary feedback, the terminal value function $V^N : \mathcal{S} \mapsto \mathbb{R}$ is given by (1) as

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

The value function at times $n = 0, \ldots, N-1$, V^n is given recursively through Bellman's equation:

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

Since the "curse of dimensionality" makes direct computation of the value function intractable, in what follows, KG will be extended to handle Bayesian classification models.

5.2. Knowledge Gradient for Binary Responses

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

Definition 2. *The knowledge gradient of measuring an alternative $\boldsymbol{x}$ while in state s is*

$$\nu^{KG}_{\boldsymbol{x}}(s) := \mathbb{E}[V^N(T(s, \boldsymbol{x}, Y)) - V^N(s)|\boldsymbol{x}, s]. \quad (9)$$

Since the outcome Y of an experiment where we made choice $\boldsymbol{x}$ is not known at the time of selection, the expectation is computed conditional on the current model specified by $s = (\boldsymbol{m}, \boldsymbol{\Sigma})$. Specifically, in the case of stochastic binary feedbacks, given a state $s = (\boldsymbol{m}, \boldsymbol{\Sigma})$, the label y for an alternative $\boldsymbol{x}$ follows from a Bernoulli distribution with a predictive distribution

$$\begin{aligned} \Pr(y = +1|\boldsymbol{x}, s) &= \int \Pr(y = +1|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w}|s) \mathrm{d}\boldsymbol{w} \\ &= \int \sigma(\boldsymbol{w}^T \boldsymbol{x}) p(\boldsymbol{w}|s) \mathrm{d}\boldsymbol{w}. \end{aligned} \quad (10)$$

We can calculate the expected value in the next state as

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

The knowledge gradient policy suggests at each time n selecting the alternative that maximizes $\nu^{\mathrm{KG}}_{\boldsymbol{x}}(S^n)$ where ties are broken randomly. The knowledge gradient policy can work with any choice of link function $\sigma(\cdot)$ and approximation procedures by adjusting the transition function $T(s, \boldsymbol{x}, \cdot)$ accordingly.

The predictive distribution $\Pr(y = +1|\boldsymbol{x}, s)$ is obtained by marginalizing under current belief $p(\boldsymbol{w}|s) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{\Sigma})$. Denoting $a = \boldsymbol{w}^T \boldsymbol{x}$ and $\delta(\cdot)$ as the Dirac delta function, we have

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

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 $p(\boldsymbol{w}|s) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{\Sigma})$ is Gaussian, we can evaluate $p(a)$ by calculating the mean and covariance (Bishop et al., 2006):

$$\mu_a = \mathbb{E}[a] = \boldsymbol{m}^T \boldsymbol{x}, \;\; \sigma_a^2 = \mathrm{Var}[a] = \boldsymbol{x}^T \boldsymbol{\Sigma} \boldsymbol{x}.$$

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

For the logistic link function, the convolution cannot be evaluated analytically. We apply the approximation $\sigma(a) \approx \Phi(\alpha a)$ with $\alpha = \pi/8$ (Barber & Bishop, 1998). Denoting $\kappa(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2}$, we have

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

Because of the one-step look ahead, the KG calculation can also benefit from the online recursive update of the belief. We summarize the KG policy with online logistic regression in Algorithm 2.

Algorithm 2 Knowledge Gradient Policy under online Bayesian Logistic Regression

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

The knowledge gradient for offline learning extends easily to bandit settings (Ryzhov et al., 2012) with the goal to minimize the cumulative regret by selecting $X^{\mathrm{KG},n}$ at each time step n as:

$$X^{\mathrm{KG},n}(S^n) = \arg\max_{\boldsymbol{x}} \Pr(y = +1|\boldsymbol{x}, S^n) + (N - n)\nu^{\mathrm{KG}}_{\boldsymbol{x}}(S^n).$$

We close this section by presenting the following finite-time bound on the MSE of the estimated weight for

Bayesian logistic regression with the proof in the supplement. Since the learner plays an active tole in selecting the measurements, the bound does not make the i.i.d. assumption of the examples. Without loss of generality, we assume $\|\boldsymbol{x}\|_2 \leq 1, \forall \boldsymbol{x} \in \mathcal{X}$.

Theorem 1. *Let $\mathcal{D}^n$ be the n measurements produced by the KG policy and $\boldsymbol{w}^n = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}|\boldsymbol{m}^0, \Sigma^0)$ with the prior distribution $Pr(\boldsymbol{w}^*) = \mathcal{N}(\boldsymbol{w}^*|\boldsymbol{m}^0, \boldsymbol{\Sigma}^0)$. Then with probability $P_d(M)$, the expected error of $\boldsymbol{w}^n$ is bounded as*

$$\mathbb{E}_{\boldsymbol{y}\sim\mathcal{B}(\mathcal{D}^n,\boldsymbol{w}^*)}||\boldsymbol{w}^n - \boldsymbol{w}^*||_2 \leq \frac{C_{min} + \lambda_{min}\left(\boldsymbol{\Sigma}^{-1}\right)}{2},$$

where the distribution $\mathcal{B}(\mathcal{D}^n, \boldsymbol{w}^*)$ is the vector on-Bernoulli distribution with $Pr(y^i = +1) = \sigma(\boldsymbol{w}^{*T}\boldsymbol{x}^i)$ of each dimension, $P_d(M)$ is the probability of a d-dimensional standard normal random variable appears in the ball with radius $M = \frac{1}{8}\frac{\lambda^2_{min}}{\sqrt{\lambda_{max}}}$ and $C_{min} = \lambda_{min}\left(\frac{1}{n}\sum_{i=1}\sigma(\boldsymbol{w}^{*T}\boldsymbol{x}^i)\big(1-\sigma(\boldsymbol{w}^{*T}\boldsymbol{x}^i)\big)\boldsymbol{x}^i(\boldsymbol{x}^i)^T\right)$.

In the special case where $\boldsymbol{\Sigma}^0 = \lambda^{-1}\boldsymbol{I}$, we have $\lambda_{max} = \lambda_{min} = \lambda$ and $M = \frac{\lambda^{3/2}}{8}$. The bound holds with higher probability $P_d(M)$ with larger λ. This is natural since a larger λ represents a normal distribution with narrower bandwidth, resulting in a more concentrated $\boldsymbol{w}^*$ around $\boldsymbol{m}^0$.

6. Experimental Results

We evaluate the proposed method on 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 $\sigma(w_0^* + \sum_{j=1}^d w_d^* x_d)$. 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.

6.1. Behavior of the KG Policy

To better understand the behavior of the KG policy, we provide snapshots of the KG policy at each iteration on a 2-dimensional synthetic dataset and a 3-dimensional synthetic dataset in one run. Fig. 1 shows the snapshot on a 2-dimensional dataset with 200 alternatives. The scatter plots illustrate the KG values with both the color and the size of the point reflecting the KG value of each 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.

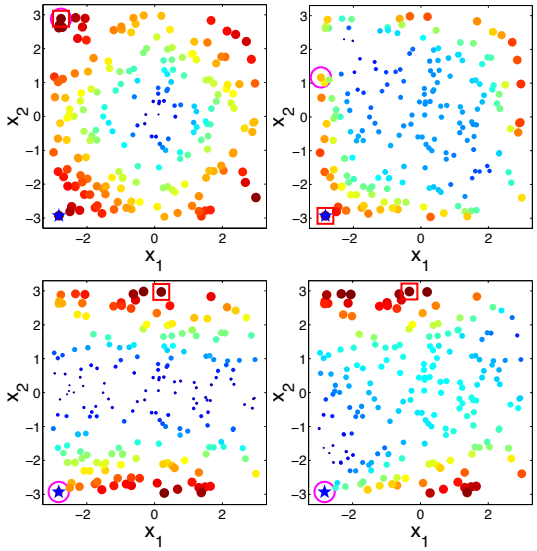


Figure 1. Snapshots on a 2-dimensional dataset. The scatter plots illustrate the KG values at 1-4 iterations from left to right, top to bottom. The star, the red square and pink circle indicate the true best alternative, the alternative to be selected and the implementation decision, respectively.

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. 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 in (Zhang & Oles, 2000). For an extreme example, when $\boldsymbol{x} = \boldsymbol{0}$ the label is always completely random for any $\boldsymbol{w}$ since $\Pr(y = +1|\boldsymbol{w}, \boldsymbol{0}) \equiv 0.5$. 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. 2 illustrates the snapshots of the KG policy on a 3-dimensional synthetic dataset with 300 alternatives. It can

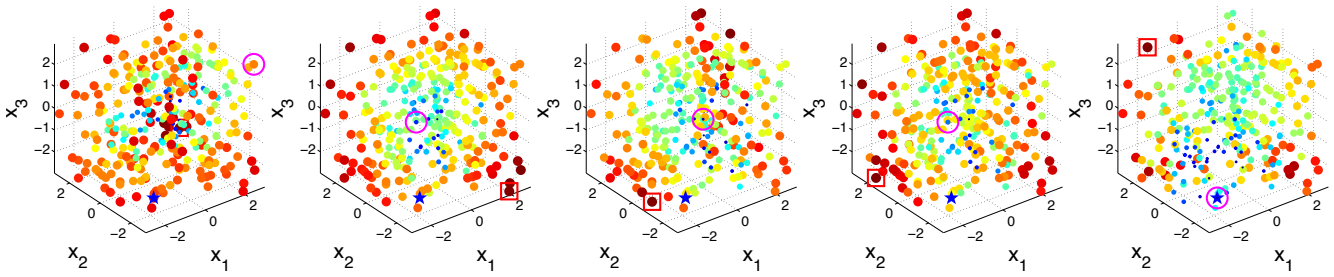
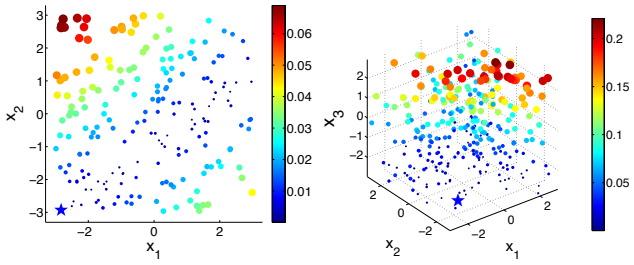


Figure 2. Snapshots on a 3-dimensional dataset. The scatter plots illustrate the KG values at 2,4,6,8,10 iterations from left to right. The star, the red square and pink circle indicate the best alternative, the alternative to be selected and the implementation decision.

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.

Also depicted in Fig. 3 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 on the 2-dimensional dateset after 4 iterations in Fig. 3(a) and on the 3-dimensional dateset after 10 iterations in Fig. 3(b). We see that in both cases, the probability at the true maximum is well approximated, while moderate error in the estimate is located away from this region of interest.



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

Figure 3. Absolute error.

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 $\boldsymbol{x}^{N+1} := \arg\max_{\boldsymbol{x}} \Pr(y = +1|\boldsymbol{x}, \boldsymbol{w}^N)$ compared to the true maximal response under weight $\boldsymbol{w}^*$:

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

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 & 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 & Li, 2011). Besides, as upper confidence bounds (UCB) methods are often considered 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 4, while different policies provides different rules for measurement decisions at each iteration. The experimental results are shown in figure 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 Fig. 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 need to learn the most in small budgets. MostUncertain and Disc perform quite well on some datasets while badly on others. A possible explanation is that the goal of active leaning 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 & Li, 2011). Yet, KG has a better performance especially during the early iterations. In the case when an

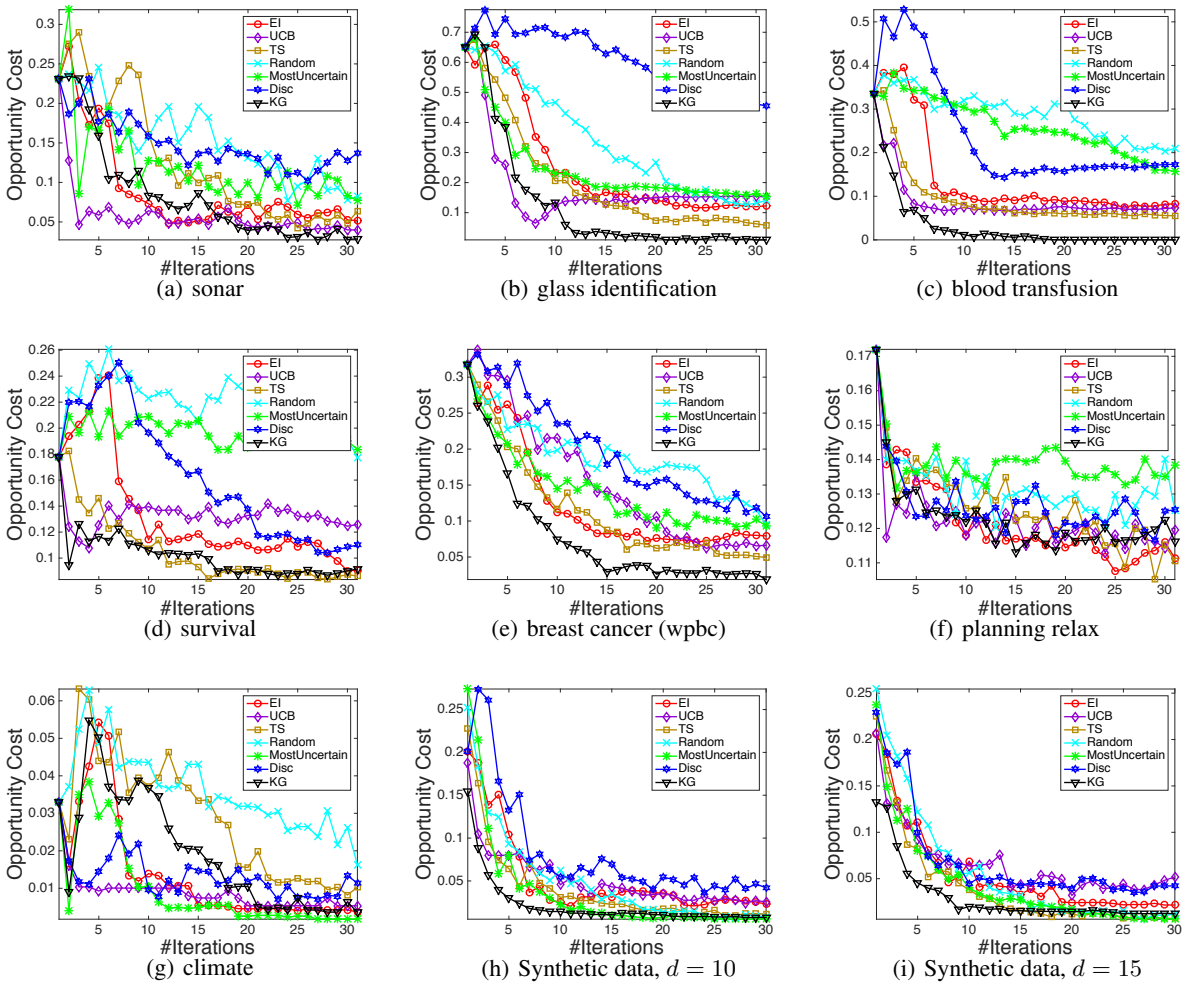


Figure 4. Opportunity cost on UCI and synthetic datasets.

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 Fig. 4(b), 4(c) and 4(e). Although both KG and EI work with the expected value of information, when EI decides which alternative to measure, it ignores the potential change of the posterior distribution resulting from the next (stochastic) outcome y.

Finally, KG, EI and TS 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 TS make decisions in the underlying binary outcome probability space and find the right balance of exploration and exploitation.

7. Conclusion

In this paper, we consider sequential decision making problems with binary outcomes where we have to run expensive experiments, forcing us to learn the most from each experiment. With a small budget of measurements, the goal is to identify the alternative with the highest probability of success as quickly as possible. Due to the sequential nature of this problem, we develop a fast online Bayesian linear classifier for general response functions. We propose a knowledge gradient policy using Bayesian linear classification belief models, for which we develope an approximation method to overcome computational challenges in finding the knowledge gradient. Other than a focus on offline optimization, we extend the knowledge gradient policy to bandit settings to minimize regret. We provide a finite-time analysis on the estimated error, and report the results of a series of experiments that demonstrate its efficiency.

Acknowledgements

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.

References

Audibert, Jean-Yves, Bubeck, Sébastien, et al. Best arm identification in multi-armed bandits. *COLT 2010-Proceedings*, 2010.

Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Barber, David and Bishop, Christopher M. Ensemble learning for multi-layer networks. *Advances in neural information processing systems*, pp. 395–401, 1998.

Bishop, Christopher M et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

Bubeck, Sébastien and Cesa-Bianchi, Nicolò. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Chapelle, Olivier and Li, Lihong. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Chick, Stephen E. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49(5):732–743, 2001.

DeGroot, M. H. *Optimal Statistical Decisions*. McGraw-Hill, 1970.

Filippi, Sarah, Cappe, Olivier, Garivier, Aurélien, and Szepesvári, Csaba. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2010.

Frazier, Peter I, Powell, Warren B, and Dayanik, Savas. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

Freund, Yoav, Seung, H Sebastian, Shamir, Eli, and Tishby, Naftali. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

Gao, Tianshi and Koller, Daphne. Active classification based on value of classifier. In *Advances in Neural Information Processing Systems*, pp. 1062–1070, 2011.

Guo, Yuhong and Schuurmans, Dale. Discriminative batch mode active learning. In *Advances in neural information processing systems*, pp. 593–600, 2008.

Gutmann, H-M. A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3): 201–227, 2001.

He, Donghai, Chick, Stephen E, and Chen, Chun-Hung. 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, 2007.

He, He, Eisner, Jason, and Daume, Hal. Imitation learning by coaching. In *Advances in Neural Information Processing Systems*, pp. 3149–3157, 2012.

Hennig, Philipp and Schuler, Christian J. Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13(1):1809–1837, 2012.

Hoffman, Matthew D, Shahriari, Bobak, and de Freitas, Nando. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *AISTATS*, pp. 365–374, 2014.

Hosmer Jr, David W and Lemeshow, Stanley. *Applied logistic regression*. John Wiley & Sons, 2004.

Huang, Deng, Allen, Theodore T, Notz, William I, and Zeng, N. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466, 2006.

Jones, Donald R. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

Jones, Donald R, Schonlau, Matthias, and Welch, William J. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13 (4):455–492, 1998.

Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Mahajan, Dhruv Kumar, Rastogi, Rajeev, Tiwari, Charu, and Mitra, Adway. Logucb: an explore-exploit algorithm for comments recommendation. In *Proceedings of*

the 21st ACM international conference on Information and knowledge management, pp. 6–15. ACM, 2012.

Mes, Martijn RK, Powell, Warren B, and Frazier, Peter I. Hierarchical knowledge gradient for sequential sampling. *The Journal of Machine Learning Research*, 12:2931–2974, 2011.

Montgomery, D. C. *Design and Analysis of Experiments*. John Wiley and Sons, 2008.

Negoescu, Diana M, Frazier, Peter I, and Powell, Warren B. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

Powell, Warren B and Ryzhov, Ilya O. *Optimal learning*. John Wiley & Sons, 2012.

Regis, Rommel G and Shoemaker, Christine A. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171, 2005.

Ryzhov, Ilya O, Powell, Warren B, and Frazier, Peter I. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1): 180–195, 2012.

Schein, Andrew I and Ungar, Lyle H. Active learning for logistic regression: an evaluation. *Machine Learning*, 68 (3):235–265, 2007.

Settles, Burr. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

Tesch, Matthew, Schneider, Jeff, and Choset, Howie. Expensive function optimization with stochastic binary outcomes. In *Proceedings of The 30th International Conference on Machine Learning*, pp. 1283–1291, 2013.

Tong, Simon and Koller, Daphne. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

Wang, Yingfei, Reyes, Kristofer G, Brown, Keith A, Mirkin, Chad A, and Powell, Warren B. 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, 2015.

Wetherill, G. B. and Glazebrook, K. D. *Sequential Methods in Statistics*. Chapman and Hall, 1986.

Wright, Stephen J and Nocedal, Jorge. *Numerical optimization*, volume 2. Springer New York, 1999.

Zhang, Tong and Oles, F. The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning,(Langley, P., ed.)*, pp. 1191–1198. Citeseer, 2000.