Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Learning to Optimize via Information-Directed Sampling

Daniel Russo, Benjamin Van Roy

To cite this article:
Daniel Russo, Benjamin Van Roy (2018) Learning to Optimize via Information-Directed Sampling. Operations Research 66(1):230-252. https://doi.org/10.1287/opre.2017.1663

http://pubsonline.informs.org/journal/opre/

OPERATIONS RESEARCH
Vol. 66, No. 1, January–February 2018, pp. 230–252
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Learning to Optimize via Information-Directed Sampling

Daniel Russo,[a] **Benjamin Van Roy**[b]

[a] Graduate School of Business, Columbia University, New York, New York 10027; [b] Stanford University, Stanford, California 94305
Contact: djr2174@gsb.columbia.edu, http://orcid.org/0000-0001-5926-8624 (DR); bvr@stanford.edu (BVR)

Received: July 17, 2014
Revised: August 12, 2016
Accepted: March 31, 2017
Published Online in Articles in Advance: October 26, 2017

Subject Classifications: decision analysis: sequential; statistics: Bayesian
Area of Review: Stochastic Models

https://doi.org/10.1287/opre.2017.1663

Abstract. We propose *information-directed sampling*—a new approach to online optimization problems in which a decision maker must balance between exploration and exploitation while learning from partial feedback. Each action is sampled in a manner that minimizes the ratio between squared expected single-period regret and a measure of information gain: the mutual information between the optimal action and the next observation.

We establish an expected regret bound for information-directed sampling that applies across a very general class of models and scales with the entropy of the optimal action distribution. We illustrate through simple analytic examples how information-directed sampling accounts for kinds of information that alternative approaches do not adequately address and that this can lead to dramatic performance gains. For the widely studied Bernoulli, Gaussian, and linear bandit problems, we demonstrate state-of-the-art simulation performance.

Funding: This work was generously supported by a research grant from Boeing, a Marketing Research Award from Adobe, and the Burt and Deedee McMurty Stanford Graduate Fellowship.
Supplemental Material: The electronic companion is available at https://doi.org/10.1287/opre.2017.1663.

Keywords: online optimization • multi-armed bandit • exploration/exploitation • information theory.

1. Introduction

In the classical multi-armed bandit problem, a decision maker repeatedly chooses from among a finite set of actions. Each action generates a random reward drawn independently from a probability distribution associated with the action. The decision maker is uncertain about these reward distributions but learns about them as rewards are observed. Strong performance requires striking a balance between *exploring* poorly understood actions and *exploiting* previously acquired knowledge to attain high rewards. Because selecting one action generates no information pertinent to other actions, effective algorithms must sample every action many times.

There has been significant interest in addressing problems with more complex *information structures*, in which sampling one action can inform the decision-maker's assessment of other actions. Effective algorithms must take advantage of the information structure to learn more efficiently. The most popular approaches to such problems extend *upper-confidence-bound* (UCB) algorithms and *Thompson sampling*, which were originally devised for the classical multi-armed bandit problem. In some cases, such as the linear bandit problem, strong performance guarantees have been established for these approaches. For some problem classes, compelling empirical results have also been presented for UCB algorithms and Thompson sampling, as well as the knowledge gradient algorithm. However, as we will demonstrate through simple analytic examples, these approaches can perform very poorly when faced with more complex information structures. Shortcomings stem from the fact that they do not adequately account for particular kinds of information.

In this paper, we propose a new approach—*information-directed sampling* (IDS)—that is designed to address this. IDS quantifies the amount learned by selecting an action through an information-theoretic measure: the mutual information between the true optimal action and the next observation. Each action is sampled in a manner that minimizes the ratio between squared expected single-period regret and this measure of information gain. Through this information measure, IDS accounts for kinds of information that alternatives fail to address.

As we will demonstrate through simple analytic examples, IDS can dramatically outperform UCB algorithms, Thompson sampling, and the knowledge-gradient algorithm. Further, by leveraging the tools of our recent information theoretic analysis of Thompson sampling (Russo and Van Roy 2016), we establish

an expected regret bound for IDS that applies across a very general class of models and scales with the entropy of the optimal action distribution. We also specialize this bound to several classes of online optimization problems, including problems with full feedback, linear optimization problems with bandit feedback, and combinatorial problems with semi-bandit feedback, in each case establishing that bounds are order optimal up to a polylogarithmic factor.

We benchmark the performance of IDS through simulations of the widely studied Bernoulli, Gaussian, and linear bandit problems, for which UCB algorithms and Thompson sampling are known to be very effective. We find that even in these settings, IDS outperforms UCB algorithms and Thompson sampling. This is particularly surprising for Bernoulli bandit problems, where UCB algorithms and Thompson sampling are known to be asymptotically optimal in the sense proposed by Lai and Robbins (1985).

IDS solves a single-period optimization problem as a proxy to an intractable multi-period problem. Solving this single-period problem may still be computationally demanding. We develop numerical methods for particular classes of online optimization problems. In some cases, our numerical methods do not compute exact or near-exact solutions but generate efficient approximations that are intended to capture key benefits of IDS. There is much more work to be done to design efficient algorithms for various problem classes and we hope that our analysis and initial collection of numerical methods will provide a foundation for further developments.

It is worth noting that the problem formulation we work with, which is presented in Section 3, is very general, encompassing not only problems with bandit feedback, but also a broad array of information structures for which observations can offer information about rewards of arbitrary subsets of actions or factors that influence these rewards. Because IDS and our analysis accommodate this level of generality, they can be specialized to problems that in the past have been studied individually, such as those involving pricing and assortment optimization (see, e.g., Rusmevichientong et al. 2010, Broder and Rusmevichientong 2012, Sauré and Zeevi 2013), though in each case, developing a computationally efficient version of IDS may require innovation.

2. Literature Review

UCB algorithms are the primary approach considered in the segment of the stochastic multi-armed bandit literature that treats problems with dependent arms. UCB algorithms have been applied to problems where the mapping from action to expected reward is a linear (Dani et al. 2008, Rusmevichientong and Tsitsiklis 2010, Abbasi-Yadkori et al. 2011), generalized linear (Filippi et al. 2010), or sparse linear (Abbasi-Yadkori et al. 2012) model; is sampled from a Gaussian process (Srinivas et al. 2012) or has small norm in a reproducing kernel Hilbert space (Srinivas et al. 2012, Valko et al. 2013b); or is a smooth (e.g., Lipschitz continuous) model (Kleinberg et al. 2008, Bubeck et al. 2011, Valko et al. 2013a). Recently, an algorithm known as Thompson sampling has received a great deal of interest. Agrawal and Goyal (2013b) provided the first analysis for linear contextual bandit problems. Russo and Van Roy (2013, 2014b) consider a more general class of models, and show that standard analysis of upper confidence bound algorithms leads to bounds on the expected regret of Thompson sampling. Very recent work of Gopalan et al. (2014) provides asymptotic frequentist bounds on the growth rate of regret for problems with dependent arms. Both UCB algorithms and Thompson sampling have been applied to other types of problems, like reinforcement learning (Jaksch et al. 2010, Osband et al. 2013) and Monte Carlo tree search (Kocsis and Szepesvári 2006, Bai et al. 2013).

In one of the first papers on multi-armed bandit problems with dependent arms, Agrawal et al. (1989a) consider a general model in which the reward distribution associated with each action depends on a common unknown parameter. When the parameter space is finite, they provide a lower bound on the asymptotic growth rate of the regret of any admissible policy as the time horizon tends to infinity and show that this bound is attainable. These results were later extended by Agrawal et al. (1989b) and Graves and Lai (1997) to apply to the adaptive control of Markov chains and to problems with infinite parameter spaces. These papers provide results of fundamental importance, but seem to have been overlooked by much of the recent literature.

Though the use of mutual information to guide sampling has been the subject of much research, dating back to the work of Lindley (1956), to our knowledge, only two other papers (Villemonteix et al. 2009, Hennig and Schuler 2012) have used the mutual information between the optimal action and the next observation to guide action selection. Each focuses on optimization of expensive-to-evaluate black-box functions. Here, *black box* indicates the absence of strong structural assumptions such as convexity and that the algorithm only has access to function evaluations, while *expensive-to-evaluate* indicates that the cost of evaluation warrants investing considerable effort to determine where to evaluate. These papers focus on settings with low-dimensional continuous action spaces, and with a Gaussian process prior over the objective function, reflecting the belief that "smoother" objective functions are more plausible than others. This approach is often called "Bayesian optimization" in the machine

learning community (Brochu et al. 2009). Both Villemonteix et al. (2009) and Hennig and Schuler (2012) propose selecting each sample to maximize the mutual information between the next observation and the optimal solution. Several papers (Hernández-Lobato et al. 2014, 2015, 2017) have extended this line of work since an initial version of our paper appeared online. The numerical routines in these papers use approximations to mutual information and may give insight into how to design efficient computational approximations to IDS.

Several features distinguish our work from that of Villemonteix et al. (2009) and Hennig and Schuler (2012). First, these papers focus on pure exploration problems: the objective is simply to learn about the optimal solution—not to attain high cumulative reward. Second, and more importantly, they focus only on problems with Gaussian process priors and continuous action spaces. For such problems, simpler approaches like UCB algorithms (Srinivas et al. 2012), probability of improvement (Kushner 1964), and expected improvement (Mockus et al. 1978) are already extremely effective. As noted by Brochu et al. (2009, p. 11), each of these algorithms simply chooses points with "*potentially* high values of the objective function: whether because the prediction is high, the uncertainty is great, or both." By contrast, a major motivation of our work is that a richer information measure is needed to address problems with more complicated information structures. Finally, we provide a variety of general theoretical guarantees for information-directed sampling, whereas Villemonteix et al. (2009) and Hennig and Schuler (2012) propose their algorithms as heuristics without guarantees. Section 9.1 shows that our theoretical guarantees extend to pure exploration problems.

The knowledge gradient (KG) algorithm uses a different measure of information to guide action selection: the algorithm computes the impact of a single observation on the quality of the decision made by a *greedy* algorithm, which simply selects the action with highest posterior expected reward. This measure was proposed by Mockus et al. (1978) and studied further by Frazier et al. (2008) and Ryzhov et al. (2012). KG seems natural since it explicitly seeks information that improves decision quality. Computational studies suggest that for problems with Gaussian priors, Gaussian rewards, and relatively short time horizons, KG performs very well. However, there are no general guarantees for KG, and even in some simple settings, it may not converge to optimality. In fact, it may select a suboptimal action in *every* period, even as the time horizon tends to infinity. IDS also measures the information provided by a single observation, but our results imply it converges to optimality. KG is discussed further in Section 4.3.3.

Our work also connects to a much larger literature on Bayesian experimental design (see Chaloner et al. 1995, for a review). Contal et al. (2014) study problems with Gaussian process priors and a method that guides exploration using the mutual information between the objective function and the next observation. This work provides a regret bound, though, as the authors' erratum indicates, the proof of the regret bound is incorrect. Recent work has demonstrated the effectiveness of *greedy* or *myopic* policies that always maximize a measure of the information gain from the next sample. Jedynak et al. (2012) and Waeber et al. (2013) consider problem settings in which this greedy policy is optimal. Another recent line of work (Golovin et al. 2010, Golovin and Krause 2011) shows that measures of information gain sometimes satisfy a decreasing returns property known as adaptive submodularity, implying the greedy policy is competitive with the optimal policy. Our algorithm also only considers the information gain because of the *next sample*, even though the goal is to acquire information over many periods. Our results establish that the manner in which IDS encourages information gain leads to an effective algorithm, even for the different objective of maximizing cumulative reward.

Finally, our work connects to the literature on partial monitoring. First introduced by (Piccolboni and Schindelhauer 2001) the partial monitoring problem encompasses a broad range of online optimization problems with limited or partial feedback. Recent work (Bartók et al. 2014) has focused on classifying the minimax-optimal scaling of regret in the problem's time horizon as a function of the level of feedback the agent receives. That work focuses most attention on cases where the agent receives very restrictive feedback, and in particular, cannot observe the reward their action generates. Our work also allows the agent to observe rich forms of feedback in response to actions they select, but we focus on a more standard decision-theoretic framework in which the agent associates their observations with a reward as specified by a utility function.

The literature we have discussed primarily focuses on contexts where the goal is to converge on an optimal action in a manner that limits exploration costs. Such methods are not geared toward problems where time preference plays an important role. A notable exception is the KG algorithm, which takes a discount factor as input to account for time preference. Francetich and Kreps (2016a, b) discuss a variety of heuristics for the discounted problem. Recent work (Russo et al. 2017) generalizes Thompson sampling to address discounted problems. We believe that IDS can also be extended to treat discounted problems, though we do not pursue that in this paper.

The regret bounds we will present build on our information-theoretic analysis of Thompson sampling (Russo and Van Roy 2016), which can be used to bound the regret of any policy in terms of its information ratio.

The information ratio of IDS is always smaller than that of TS, and therefore, bounds on the information ratio of TS provided in Russo and Van Roy (2016) yield regret bounds for IDS. This observation and a preliminary version of our results was first presented in a conference paper (Russo and Van Roy 2014a). Recent work by Bubeck et al. (2015) and Bubeck and Eldan (2016) build on ideas from (Russo and Van Roy 2016) in another direction by bounding the information ratio when the reward function is convex and using that bound to study the order of regret in adversarial bandit convex optimization.

3. Problem Formulation

We consider a general probabilistic, or Bayesian, formulation in which uncertain quantities are modeled as random variables. The decision maker sequentially chooses actions $(A_t)_{t\in\mathbb{N}}$ from a finite action set $\mathcal{A}$ and observes the corresponding outcomes $(Y_{t,A_t})_{t\in\mathbb{N}}$. There is a random outcome $Y_{t,a} \in \mathcal{Y}$ associated with each action $a \in \mathcal{A}$ and time $t \in \mathbb{N}$. Let $Y_t \equiv (Y_{t,a})_{a\in\mathcal{A}}$ be the vector of outcomes at time $t \in \mathbb{N}$. There is a random variable θ such that, conditioned on θ, $(Y_t)_{t\in\mathbb{N}}$ is an iid sequence. This can be thought of as a Bayesian formulation, where randomness in θ captures the decision-maker's prior uncertainty about the true nature of the system, and the remaining randomness in Y_t captures idiosyncratic randomness in observed outcomes.

The agent associates a reward $R(y)$ with each outcome $y \in \mathcal{Y}$, where the reward function $R\colon \mathcal{Y} \to \mathbb{R}$ is fixed and known. Let $R_{t,a} = R(Y_{t,a})$ denote the realized reward of action a at time t. Uncertainty about θ induces uncertainty about the true optimal action, which we denote by $A^* \in \arg\max_{a\in\mathcal{A}} \mathbb{E}[R_{1,a} \mid \theta]$. The T–period *regret* of the sequence of actions $A_1, \ldots, A_T$ is the random variable

$$\text{Regret}(T) := \sum_{t=1}^{T} (R_{t,A^*} - R_{t,A_t}), \tag{1}$$

which measures the cumulative difference between the reward earned by an algorithm that always chooses the optimal action and actual accumulated reward up to time T. In this paper we study expected regret

$$\mathbb{E}[\text{Regret}(T)] = \mathbb{E}\left[\sum_{t=1}^{T} (R_{t,A^*} - R_{t,A_t})\right], \tag{2}$$

where the expectation is taken over the randomness in the actions A_t and the outcomes Y_t, and over the prior distribution over θ. This measure of performance is commonly called *Bayesian regret* or *Bayes risk*.

The action A_t is chosen based on the history of observations $\mathcal{F}_t = (A_1, Y_{1,A_1}, \ldots, A_{t-1}, Y_{t-1,A_{t-1}})$ up to time t. The action A_t is chosen based on the history of observations $\mathcal{F}_t = (A_1, Y_{1,A_1}, \ldots, A_{t-1}, Y_{t-1,A_{t-1}})$ up to time t. Formally, a *randomized policy* $\pi = (\pi_t)_{t\in\mathbb{N}}$ is a sequence of deterministic functions, where $\pi_t(\mathcal{F}_t)$ specifies a probability distribution over the action set $\mathcal{A}$. Let $\mathcal{D}(\mathcal{A})$ denote the set of probability distributions over $\mathcal{A}$. The action A_t is a selected by sampling independently from $\pi_t(\mathcal{F}_t)$. With some abuse of notation, we will typically write this distribution as π_t, where $\pi_t(a) = \mathbb{P}(A_t = a \mid \mathcal{F}_t)$ denotes the probability assigned to action a given the observed history. We explicitly display the dependence of regret on the policy π, letting $\mathbb{E}[\text{Regret}(T, \pi)]$ denote the expected value given by (2) when the actions $(A_1, \ldots, A_T)$ are chosen according to π.

Further Notation. Set $\alpha_t(a) = \mathbb{P}(A^* = a \mid \mathcal{F}_t)$ to be the posterior distribution of A^*. For two probability measures P and Q over a common measurable space, if P is absolutely continuous with respect to Q, the *Kullback-Leibler divergence* between P and Q is

$$D_{\text{KL}}(P||Q) = \int \log\left(\frac{dP}{dQ}\right) dP \tag{3}$$

where dP/dQ is the Radon-Nikodym derivative of P with respect to Q. For a probability distribution P over a finite set $\mathcal{X}$, the *Shannon entropy* of P is defined as $H(P) = -\sum_{x\in\mathcal{X}} P(x)\log(P(x))$. The *mutual information* under the posterior distribution between two random variables $X_1\colon \Omega \to \mathcal{X}_1$, and $X_2\colon \Omega \to \mathcal{X}_2$, denoted by

$$\begin{aligned} &I_t(X_1; X_2) \\ &:= D_{\text{KL}}\left[\mathbb{P}((X_1, X_2) \in \cdot \mid \mathcal{F}_t) \| \mathbb{P}(X_1 \in \cdot \mid \mathcal{F}_t)\mathbb{P}(X_2 \in \cdot \mid \mathcal{F}_t)\right], \end{aligned} \tag{4}$$

is the Kullback-Leibler divergence between the joint posterior distribution of X_1 and X_2 and the product of the marginal distributions. Note that $I_t(X_1; X_2)$ is a random variable because of its dependence on the conditional probability measure $\mathbb{P}(\cdot \mid \mathcal{F}_t)$.

To reduce notation, we define the *information gain* from an action a to be $g_t(a) := I_t(A^*; Y_{t,a})$. As shown, for example, in Gray (2011, Lemma 5.5.6), this is equal to the expected reduction in entropy of the posterior distribution of A^* because of observing $Y_t(a)$:

$$g_t(a) = \mathbb{E}[H(\alpha_t) - H(\alpha_{t+1}) \mid \mathcal{F}_t, A_t = a], \tag{5}$$

which plays a crucial role in our results. Let $\Delta_t(a) := \mathbb{E}[R_{t,A^*} - R_{t,a} \mid \mathcal{F}_t]$ denote the expected instantaneous regret of action a at time t.

We use overloaded notation for $g_t(\cdot)$ and $\Delta_t(\cdot)$. For an action sampling distribution $\pi \in \mathcal{D}(\mathcal{A})$, $g_t(\pi) := \sum_{a\in\mathcal{A}} \pi(a) g_t(a)$ denotes the expected information gain when actions are selected according to π, and $\Delta_t(\pi) = \sum_{a\in\mathcal{A}} \pi(a)\Delta_t(a)$ is defined analogously. Finally, we sometimes use the shorthand notation $\mathbb{E}_t[\cdot] = \mathbb{E}_t[\cdot \mid \mathcal{F}_t]$ for conditional expectations under the posterior distribution, and similarly write $\mathbb{P}_t(\cdot) = \mathbb{P}(\cdot \mid \mathcal{F}_t)$.

4. Algorithm Design Principles

The primary contribution of this paper is IDS, a general principle for designing action-selection algorithms. We will define IDS in this section, after discussing motivations underlying its structure. Further, through a set of examples, we will illustrate how alternative design principles fail to account for particular kinds of information and therefore can be dramatically outperformed by IDS.

4.1. Motivation

Our goal is to minimize expected regret over a time horizon T. This is achieved by a *Bayes-optimal* policy, which, in principle, can be computed via dynamic programming. Unfortunately, computing, or even storing, this Bayes-optimal policy is generally infeasible. For this reason, there has been significant interest in developing computationally efficient heuristics.

As with much of the literature, we are motivated by contexts where the time horizon T is "large." For large T and moderate times $t \ll T$, the mapping from belief state to action prescribed by the Bayes-optimal policy does not vary significantly from one time period to the next. As such, it is reasonable to restrict attention to stationary heuristic policies. IDS falls in this category.

IDS is motivated largely by a desire to overcome shortcomings of currently popular design principles. In particular, it accounts for kinds of information that alternatives fail to adequately address:

1. *Indirect information.* IDS can select an action to obtain useful feedback about other actions even if there will be no useful feedback about the selected action.

2. *Cumulating information.* IDS can select an action to obtain feedback that does not immediately enable higher expected reward but can eventually do so when combined with feedback from subsequent actions.

3. *Irrelevant information.* IDS avoids investments in acquiring information that will not help to determine which actions ought to be selected.

Examples presented in Section 4.3 aim to contrast the manner in which IDS and alternative approaches treat these kinds of information.

It is worth noting that we refer to IDS as a *design principle* rather than an *algorithm*. The reason is that IDS does not specify steps to be carried out in terms of basic computational operations but only an abstract objective to be optimized. As we will discuss later, for many problem classes of interest, like the Bernoulli bandit, the Gaussian bandit, and the linear bandit, one can develop tractable algorithms that implement IDS. The situation is similar for upper confidence bounds, Thompson sampling, expected improvement maximization, and knowledge gradient; these are abstract design principles that lead to tractable algorithms for specific problem classes.

4.2. Information-Directed Sampling

IDS balances between obtaining low expected regret in the current period and acquiring new information about which action is optimal. It does this by minimizing over all action sampling distributions $\pi \in \mathcal{D}(\mathcal{A})$ the ratio between the square of expected regret $\Delta_t(\pi)^2$ and information gain $g_t(\pi)$ about the optimal action A^*. In particular, the policy $\pi^{\text{IDS}} = (\pi_1^{\text{IDS}}, \pi_2^{\text{IDS}}, \ldots)$ is defined by

$$\pi_t^{\text{IDS}} \in \arg\min_{\pi \in \mathcal{D}(\mathcal{A})} \left\{ \Psi_t(\pi) := \frac{\Delta_t(\pi)^2}{g_t(\pi)} \right\}. \tag{6}$$

We call $\Psi_t(\pi)$ the *information ratio* of an action sampling distribution π. It measures the squared regret incurred per-bit of information acquired about the optimum. IDS myopically minimizes this notion of cost-per-bit of information in each period.

Note that IDS is *stationary randomized policy*: randomized in that each action is randomly sampled from a distribution and stationary in that this action distribution is determined by the posterior distribution of θ and otherwise independent of the time period. It is natural to wonder whether randomization plays a fundamental role or if a stationary deterministic policy can offer similar behavior. The following example sheds light on this matter.

Example 1 (A Known Standard). Consider a problem with two actions $\mathcal{A} = \{a_1, a_2\}$. Rewards from a_1 are known to be distributed Bernoulli(1/2). The distribution of rewards from a_2 is Bernoulli(3/4) with prior probability p_0 and is Bernoulli(1/4) with prior probability $1 - p_0$.

Consider a stationary deterministic policy for this problem. With such a policy, each action A_t is a deterministic function of p_{t-1}, the posterior probability conditioned on observations made through period $t-1$. Suppose that for some $p_0 > 0$, the policy selects $A_1 = a_1$. Since this is an uninformative action, $p_t = p_0$ and $A_t = a_1$ for all t, and thus, expected regret grows linearly with time. If, on the other hand, $A_1 = a_2$ for all $p_0 > 0$ then $A_t = a_2$ for all t, which again results in expected regret that grows linearly with time. It follows that, for any deterministic stationary policy, there exists a prior probability p_0 such that expected regret grows linearly with time.

In Section 5, we will establish a sublinear bound on expected regret of IDS. The result implies that, when applied to the preceding example, the expected regret of IDS does not grow linearly as does that of any stationary deterministic policy. This suggests that randomization plays a fundamental role.

It may appear that the need for randomization introduces great complexity since the solution of the optimization problem (6) takes the form of a distribution over actions. However, an important property of this

problem dramatically simplifies solutions. In particular, as we will establish in Section 6, there is always a distribution with support of at most two actions that attains the minimum in (6).

4.3. Alternative Design Principles

Several alternative design principles have figured prominently in the literature. However, each of them fails to adequately address one or more of the categories of information enumerated in Section 4.1. This motivated our development of IDS. In this section, we will illustrate through a set of examples how IDS accounts for such information while alternatives fail.

4.3.1. Upper Confidence Bounds and Thompson Sampling. Upper confidence bound (UCB) and Thompson sampling (TS) are two of the most popular principles for balancing between exploration and exploitation. As data are collected, both approaches do not only estimate the rewards generated by different actions, but carefully track the uncertainty in their estimates. They then continue to experiment with all actions that could *plausibly* be optimal given the observed data. This guarantees actions are not prematurely discarded, but, in contrast to more naive approaches like the ϵ-greedy algorithm, also ensures that samples are not wasted on clearly suboptimal actions.

With a UCB algorithm, actions are selected through two steps. First, for each action $a \in \mathcal{A}$ an upper confidence bound $B_t(a)$ is constructed. Then, the algorithm selects an action $A_t \in \arg\max_{a\in\mathcal{A}} B_t(a)$ with maximal upper confidence bound. The upper confidence bound $B_t(a)$ represents the greatest mean reward value that is statistically plausible. In particular, $B_t(a)$ is typically constructed to be optimistic ($B_t(a) \geqslant \mathbb{E}[R_{t,a} \mid \theta]$) and asymptotically consistent ($B_t(a) \to \mathbb{E}[R_{t,a} \mid \theta]$ as data about the action a accumulates).

A TS algorithm simply samples each action according to the posterior probability that it is optimal. In particular, at each time t, an action is sampled from $\pi_t^{\mathrm{TS}} = \alpha_t$. This means that for each $a \in \mathcal{A}$, $\mathbb{P}(A_t = a \mid \mathcal{F}_t) = \mathbb{P}(A^* = a \mid \mathcal{F}_t) = \alpha_t(a)$. This algorithm is sometimes called *probability matching* because the action selection distribution is *matched* to the posterior distribution of the optimal action.

For some problem classes, UCB and TS lead to efficient and empirically effective algorithms with strong theoretical guarantees. Specific UCB and TS algorithms are known to be asymptotically efficient for multi-armed bandit problems with independent arms (Lai and Robbins 1985, Lai 1987, Cappé et al. 2013, Kaufmann et al. 2012b, Agrawal and Goyal 2013a) and satisfy strong regret bounds for some problems with dependent arms (Dani et al. 2008, Rusmevichientong and Tsitsiklis 2010, Srinivas et al. 2012, Bubeck et al. 2011, Filippi et al. 2010, Gopalan et al. 2014, Russo and Van Roy 2014b).

Unfortunately, as the following examples demonstrate, UCB and TS do not pursue indirect information and because of that can perform very poorly relative to IDS for some natural problem classes. A common feature of UCB and TS that leads to poor performance in these examples is that they restrict attention to sampling actions that have some chance of being optimal. This is the case with TS because each action is selected according to the probability that it is optimal. With UCB, the upper-confidence-bound of an action known to be suboptimal will always be dominated by others.

Our first example is somewhat contrived but designed to make the point transparent.

Example 2 (A Revealing Action). Let $\mathcal{A} = \{0, 1, \ldots, K\}$ consist of $K + 1$ actions and suppose that θ is drawn uniformly at random from a finite set $\Theta = \{1, \ldots, K\}$ of K possible values. Consider a problem with bandit-feedback $Y_{t,a} = R_{t,a}$. Under θ, the reward of action a is

$$R_{t,a} = \begin{cases} 1 & \theta = a, \\ 0 & \theta \neq a,\ a \neq 0, \\ 1/2\theta & a = 0. \end{cases}$$

Note that action 0 is known to never yield the maximal reward, and is therefore never selected by TS or UCB. Instead, these algorithms will select among actions $\{1, \ldots, K\}$, ruling out only a single action at a time until a reward 1 is earned and the optimal action is identified. Their expected regret therefore grows linearly in K. IDS is able to recognize that much more is learned by drawing action 0 than by selecting one of the other actions. In fact, selecting action 0 immediately identifies the optimal action. IDS selects this action, learns which action is optimal, and selects that action in all future periods. Its regret is independent of K.

Our second example may be of greater practical significance. It represents the simplest case of a sparse linear model.

Example 3 (Sparse Linear Model). Consider a linear bandit problem where $\mathcal{A} \subset \mathbb{R}^d$ and the reward from an action $a \in \mathcal{A}$ is $a^T\theta^*$. The true parameter θ is known to be drawn uniformly at random from the set of 1-sparse vectors $\Theta = \{\theta' \in \{0,1\}^d : \|\theta'\|_0 = 1\}$. For simplicity, assume $d = 2^m$ for some $m \in \mathbb{N}$. The action set is taken to be the set of vectors in $\{0,1\}^d$ normalized to be a unit vector in the L^1 norm: $\mathcal{A} = \{x/\|x\|_1 : x \in \{0,1\}^d, x \neq 0\}$.

For this problem, when an action a is selected and $y = a^T\theta \in \{0, 1/\|a\|_0\}$ is observed, each $\theta' \in \Theta$ with $a^T\theta' \neq y$ is ruled out. Let Θ_t denote the set of all parameters in Θ that are consistent with the rewards observed up to time t and let $\mathcal{I}_t = \{i \in \{1, \ldots, d\} : \theta'_i = 1, \theta' \in \Theta_t\}$ denote the corresponding set of possible positive components.

Note that $A^* = \theta$. That is, if θ were known, choosing the action θ would yield the highest possible reward.

TS and UCB algorithms only choose actions from the support of A^* and therefore will only sample actions $a \in \mathcal{A}$ that, like A^*, have only a single positive component. Unless that is also the positive component of θ, the algorithm will observe a reward of zero and rule out only one element of $\mathcal{I}_t$. In the worst case, the algorithm requires d samples to identify the optimal action.

Now, consider an application of IDS to this problem. The algorithm essentially performs binary search: it selects $a \in \mathcal{A}$ with $a_i > 0$ for half of the components $i \in \mathcal{I}_t$ and $a_i = 0$ for the other half as well as for any $i \notin \mathcal{I}_t$. After just $\log_2(d)$ time steps the true value of the parameter vector θ is identified.

To see why this is the case, first note that all parameters in Θ_t are equally likely and hence the expected reward of an action a is $(1/|\mathcal{I}_t|)\sum_{i\in I_t} a_i$. Since $a_i \geqslant 0$ and $\sum_i a_i = 1$ for each $a \in \mathcal{A}$, every action whose positive components are in I_t yields the highest possible expected reward of $1/|\mathcal{I}_t|$. Therefore, binary search minimizes expected regret in period t for this problem. At the same time, binary search is assured to rule out half of the parameter vectors in Θ_t at each time t. This is the largest possible expected reduction, and also leads to the largest possible information gain about A^*. Since binary search both minimizes expected regret in period t and uniquely maximizes expected information gain in period t, it is the sampling strategy followed by IDS.

In this setting we can explicitly calculate the information ratio of each policy, and the difference between them highlights the advantages of information-directed sampling. We have

$$\Psi_1(\pi_1^{\mathrm{TS}}) = \frac{(d-1)^2/d^2}{\log(d)/d + (d-1/d)\log(d/(d-1))} \sim \frac{d}{\log(d)},$$

$$\Psi_1(\pi_1^{\mathrm{IDS}}) = \frac{1}{\log(2)}\left(1-\frac{1}{d}\right)^2 \sim \frac{1}{\log(2)},$$

where $h(d) \sim f(d)$ if $h(d)/f(d) \to 1$ as $d \to \infty$. When the dimension d is large, $\Psi_1(\pi_1^{\mathrm{IDS}})$ is much smaller.

Our final example involves an assortment optimization problem.

Example 4 (Assortment Optimization). Consider the problem of repeatedly recommending an assortment of products to a customer. The customer has unknown type $\theta \in \Theta$ where $|\Theta| = n$. Each product is geared toward customers of a particular type, and the assortment $a \in \mathcal{A} = \Theta^m$ of m products offered is characterized by the vector of product types $a = (a_1, \ldots, a_m)$. We model customer responses through a random utility model in which customers are more likely to derive high value from a product geared toward their type. When offered an assortment of products a, the customer associates with the ith product utility $U_{\theta,i}^{(t)}(a) = \beta \mathbf{1}_{\{a_i=\theta\}} + W_i^{(t)}$, where W_i^t follows an extreme–value distribution and $\beta \in \mathbb{R}$ is a known constant. This is a standard multinomial logit discrete choice model. The probability a customer of type θ chooses product i is given by

$$\frac{\exp\{\beta \mathbf{1}_{\{a_i=\theta\}}\}}{\sum_{j=1}^m \exp\{\beta \mathbf{1}_{\{a_j=\theta\}}\}}.$$

When an assortment a is offered at time t, the customer makes a choice $I_t = \arg\max_i U_{\theta i}^{(t)}(a)$ and leaves a review $U_{\theta I_t}^{(t)}(a)$ indicating the utility derived from the product, both of which are observed by the recommendation system. The reward to the recommendation system is the normalized utility of the customer $U_{\theta I_t}^{(t)}(a)/\beta$.

If the type θ of the customer were known, then the optimal recommendation would be $A^* = (\theta, \theta, \ldots, \theta)$, which consists only of products targeted at the customer's type. Therefore, both TS and UCB would only offer assortments consisting of a single type of product. Because of this, TS and UCB each require order n samples to learn the customer's true type. IDS will instead offer a *diverse* assortment of products to the customer, allowing it to learn much more quickly.

To render issues more transparent, suppose that θ is drawn uniformly at random from Θ and consider the behavior of each type of algorithm in the limiting case where $\beta \to \infty$. In this regime, the probability a customer chooses a product of type θ if it is available tends to 1, and the normalized review $\beta^{-1}U_{\theta I_t}^{(t)}(a)$ tends to $\mathbf{1}_{\{a_{I_t}=\theta\}}$, an indicator for whether the chosen product is of type θ. The initial assortment offered by IDS will consist of m different and previously untested product types. Such an assortment maximizes both the algorithm's expected reward in the next period and the algorithm's information gain, since it has the highest probability of containing a product of type θ. The customer's response almost perfectly indicates whether one of those items was of type θ. The algorithm continues offering assortments containing m unique, untested, product types until a review near $U_{\theta I_t}^{(t)}(a) \approx \beta$ is received. With extremely high probability, this takes at most $\lceil n/m \rceil$ time periods. By diversifying the m products in the assortment, the algorithm learns a factor of m times faster.

As in the previous example, we can explicitly calculate the information ratio of each policy, and the difference between them highlights the advantages of IDS. The information ratio of IDS is more than m times smaller:

$$\Psi_1(\pi_1^{\mathrm{TS}}) = \frac{(1-1/n)^2}{(1/n)\log(n) + (n-1/n)\log(n/(n-1))} \sim \frac{n}{\log(n)},$$

$$\Psi_1(\pi_1^{\mathrm{IDS}}) = \frac{(1-m/n)^2}{(m/n)\log(n/m) + (n-m/n)\log(n/(n-m))} \leqslant \frac{n}{m\log(n/m)}.$$

4.3.2. Other Information-Directed Approaches. Another natural information-directed algorithm aims to maximize the information acquired about the uncertain model parameter θ. In particular, consider an algorithm that selects the action at time t that maximizes the weighted combination of the expected reward the action generates and the information it generates about the uncertain model parameter θ: $\mathbb{E}_t[R_{t,a}] + \lambda I_t(Y_{t,a}; \theta)$. Throughout this section, we will refer to this algorithm as θ-IDS. While such an algorithm can perform well on particular examples, the next example highlights that it may invest in acquiring information about θ that is irrelevant to the decision problem.

Example 5 (Unconstrained Assortment Optimization). Consider again the problem of repeatedly recommending assortments of products to a customer with unknown preferences. The recommendation system can choose any subset of products $a \subset \{1, \dots, n\}$ to display. When offered assortment a at time t, the customer chooses the item $J_t = \arg\max_{i \in a} \theta_i$ and leaves the review $R_{t,a} = \theta_{J_t}$, where θ_i is the utility associated with product i. The recommendation system observes both J_t and the review $R_{t,a}$ and has the goal of learning to offer the assortment that yields the best outcome for the customer and maximizes the review $R_{t,a}$. Suppose that θ is drawn as a uniformly random permutation of the vector $(1, \frac{1}{2}, \frac{1}{3}, \dots, 1/n)$. The customer is known to assign utility 1 to her most preferred item, $1/2$ to the next best item, $1/3$ to the third best, and so on, but the rank ordering is unknown.

In this example, no learning is required to offer an optimal assortment: since there is no constraint on the size of the assortment, it is always best to offer the full collection of products $a = \{1, \dots, n\}$ and allow the customer to choose the most preferred. Offering this assortment reveals which item is most preferred by the customer, but it reveals nothing about her preferences about others. When applied to this problem, θ-IDS begins by offering the full assortment $A_1 = \{1, \dots, n\}$, which yields a reward of 1, and, by revealing the top item, yields information of $I_1(Y_{1,A_1}; \theta) = \log(n)$. But, if $1/2 < \lambda \log(n-1)$, which is guaranteed for sufficiently large n, it continues experimenting with suboptimal assortments. In the second period, it will offer the assortment A_2 consisting of all products except $\arg\max_i \theta_i$. Playing this assortment reveals the customer's second most preferred item, and yields information gain $I_2(Y_{2,A_2}; \theta) = \log(n-1)$. This process continues until the first period k where $\lambda \log(n+1-k) < 1 - 1/k$.

To learn to offer effective assortments, θ-IDS tries to learn as much as possible about the customer's preferences. In doing this, the algorithm inadvertently invests experimentation effort in information that is irrelevant to choosing an optimal assortment. On the other hand, IDS recognizes that the optimal assortment $A^* = \{1, \dots, n\}$ does not depend on full knowledge of the vector θ, and therefore does not invest in identifying θ.

As shown in Section 9.2, our analysis can be adapted to provide regret bounds for a version of IDS that uses information gain with respect to θ, rather than with respect to A^*. These regret bounds depend on the entropy of θ, whereas the bound for IDS depends on the entropy of A^*, which can be much smaller.

4.3.3. Expected Improvement and the Knowledge Gradient. We now consider two algorithms that measure the quality of the best decision that can be made based on current information, and we encourage gathering observations that are expected to immediately increase this measure. The first is the expected improvement algorithm, which is one of the most widely used techniques in the active field of Bayesian optimization (see Brochu et al. 2009). Define $\mu_{t,a} = \mathbb{E}[R_{t,a} \mid \mathcal{F}_t]$ to be the expected reward generated by a under the posterior, and $V_t = \max_{a'} \mu_{t,a'}$ to be the best objective value attainable given current information. The expected improvement of action a is defined to be $\mathbb{E}[\max\{f_\theta(a), V_t\} \mid \mathcal{F}_t]$, where $f_\theta(a) = \mathbb{E}[R_{t,a} \mid \theta]$ is the expected reward generated by action a under the unknown true parameter θ. The EGO algorithm aims to identify high performing actions by sequentially sampling those that yield the highest expected improvement. Similar to UCB algorithms, this encourages the selection of actions that could potentially offer great performance. Unfortunately, like these UCB algorithms, this measure of improvement does not place value on indirect information: it will not select an action that provides useful feedback about other actions unless the mean reward of that action might exceed V_t. For example, the expected improvement algorithm cannot treat the problem described in Example 2 in a satisfactory manner.

The knowledge gradient algorithm (Ryzhov et al. 2012) uses a modified improvement measure. At time t, it computes

$$v_{t,a}^{KG} := \mathbb{E}[V_{t+1} \mid \mathcal{F}_t, A_t = a] - V_t$$

for each action a. If V_t measures the quality of decision that can be made based on current information, then $v_{t,a}^{KG}$ captures the immediate improvement in decision quality from sampling action a and observing $Y_{t,a}$. For a problem with time horizon T, the KG policy selects an action in time period t by maximizing $\mu_{t,a} + (T-t)v_{t,a}^{KG}$ over actions $a \in \mathcal{A}$.

Unlike expected improvement, the measure $v_{t,a}^{KG}$ of the value of sampling an action places value on indirect information. In particular, even if an action is known to yield low expected reward, sampling that action could lead to a significant increase in V_t by providing information about other actions. Unfortunately, there are no general guarantees for KG, and it sometimes struggles with cumulating information; individual observations

that provide information about A^* may not be immediately useful for making decisions in the next period, and therefore may lead to no improvement in V_t.

Example 1 provides one simple illustration of this phenomenon. In that example, action 1 is known to yield a mean reward of 1/2. When $p_0 \leqslant 1/4$, upon sampling action 2 and observing a reward 1, the posterior expected reward of action 2 becomes

$$\mathbb{E}[R_{2,a_2} \mid R_{1,a_2} = 1] = \frac{p_0(3/4)}{p_0(3/4) + (1-p_0)(1/4)} \leqslant 1/2.$$

In particular, a single sample could never be influential enough to change which action has the highest posterior expected reward. Therefore, $v^{KG}_{t,a_2} = 0$, and the KG decision rule selects action 1 in the first period. Since nothing is learned from the resulting observation, it will continue selecting action 1 in all subsequent periods. Even as the time horizon T tends to infinity, the KG policy would never select action 2. Its cumulative regret over T time periods is equal to $(p_0/4)T$, which grows linearly with T.

In this example, although sampling action 2 will not immediately shift the decision-maker's prediction of the best action ($\arg\max_a \mathbb{E}[\theta_a \mid \mathcal{F}_1]$), these samples influences her posterior beliefs and reduce uncertainty about which action is optimal. As a result, IDS will always assign positive probability to sampling the second action. More broadly, IDS places value on information that is pertinent to the decision problem, even if that information will not directly improve performance on its own. This is useful when one must combine multiple pieces of information in order to effectively learn.

To address problems like Example 1, Frazier and Powell (2010) propose KG*—a modified form of KG that considers the value of sampling a single action many times. This helps to address some cases— those where a single sample of an action provides no value even though sampling the action several times could be quite valuable—but this modification may not address more general problems. As shown in the next example, even for problems with independent arms, the value of perfectly observing a single action could be exactly zero, even if there is value to combining information from multiple actions.

Example 6 (Heterogeneous Priors). Consider a problem with two actions $\mathcal{A} = \{1, 2\}$. When action $a \in \mathcal{A}$ is selected, a reward of θ_a is observed, where $\theta_1 \in \{0.4, 0.6\}$ and $\theta_2 \in \{0.5, 0.7\}$. Let θ_1 and θ_2 be independent, with $\mathbb{P}(\theta_1 = 0.4) = \mathbb{P}(\theta_1 = 0.6) = 0.5$ and $\mathbb{P}(\theta_2 = 0.5) = \mathbb{P}(\theta_2 = 0.7) = 0.5$, so that $\mathbb{E}[\theta_1] = 0.5$ and $\mathbb{E}[\theta_2] = 0.6$.

In this example, $\theta_1 \leqslant \mathbb{E}[\theta_2]$ and $\mathbb{E}[\theta_1] \leqslant \theta_2$ with certainty. As such, observing either θ_1 or θ_2 alone does not enable a better decision. On the other hand, observing both θ_1 and θ_2 is valuable, since it is possible that $\theta_1 = 0.6 \geqslant 0.5 = \theta_2$. KG* does not account for this possibility and would repeatedly select action 2 since it only considers what might be learned from sampling a single action multiple times. The manner in which IDS makes use of mutual information addresses such situations; in the case of Example 6, $I(A^*; \theta_1) > 0$ and $I(A^*; \theta_2) > 0$, so mutual information indicates that sampling either action provides information about the optimum, and consequently, IDS randomizes between the two actions.

5. Regret Bounds

This section establishes regret bounds for information-directed sampling for several of the most widely studied classes of online optimization problems. These regret bounds follow from our recent information theoretic analysis of Thompson sampling (Russo and Van Roy 2016). In the next subsection, we establish a regret bound for any policy in terms of its information ratio. Because the information ratio of IDS is always smaller than that of TS, the bounds on the information ratio of TS provided in Russo and Van Roy (2016) immediately yield regret bounds for IDS for a number of important problem classes.

5.1. General Bound

We begin with a general result that bounds the regret of *any policy* in terms of its information ratio and the entropy of the optimal action distribution. Recall that we have defined the information ratio of an action sampling distribution to be $\Psi_t(\pi) := \Delta_t(\pi)^2 / g_t(\pi)$; it is the squared expected regret the algorithm incurs per-bit of information it acquires about the optimum. The entropy of the optimal action distribution $H(\alpha_1)$ captures the magnitude of the decision-maker's initial uncertainty about which action is optimal. One can then interpret the next result as a bound on regret that depends on the cost of acquiring new information and the total amount of information that needs to be acquired.

Proposition 1. *For any policy $\pi = (\pi_1, \pi_2, \pi_3, \ldots)$ and time $T \in \mathbb{N}$,*

$$\mathbb{E}[\text{Regret}(T, \pi)] \leqslant \sqrt{\bar{\Psi}_T(\pi) H(\alpha_1) T},$$

where

$$\bar{\Psi}_T(\pi) \equiv \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi[\Psi_t(\pi_t)]$$

is the average expected information ratio under π.

We will use the following immediate corollary of Proposition 1, which relies on a uniform bound on the information ratio of the form $\Psi_t(\pi_t) \leqslant \lambda$ rather than a bound on the average expected information ratio.

Corollary 1. *Fix a deterministic $\lambda \in \mathbb{R}$ and a policy $\pi = (\pi_1, \pi_2, \ldots)$ such that $\Psi_t(\pi_t) \leqslant \lambda$ almost surely for each $t \in \{1, \ldots, T\}$. Then,*

$$\mathbb{E}[\text{Regret}(T, \pi)] \leqslant \sqrt{\lambda H(\alpha_1) T}.$$

5.2. Specialized Bounds on the Minimal Information Ratio

We now establish upper bounds on the information ratio of IDS in several important settings, which yields explicit regret bounds when combined with Corollary 1. These bounds show that, in any period, the algorithm's expected regret can only be large if it is expected to acquire a lot of information about which action is optimal. In this sense, it effectively balances between exploration and exploitation in *every* period. For each problem setting, we will compare our upper bounds on expected regret with known lower bounds.

The bounds on the information ratio also help to clarify the role it plays in our results: it roughly captures the extent to which sampling some actions allows the decision maker to make inferences about *other* actions. In the worst case, the ratio depends on the number of actions, reflecting the fact that actions could provide no information about others. For problems with full information, the information ratio is bounded by a numerical constant, reflecting that sampling one action perfectly reveals the rewards that would have been earned by selecting any other action. The problems of online linear optimization under "bandit feedback" and under "semi-bandit feedback" lie between these two extremes, and the ratio provides a natural measure of each problem's information structure. In each case, our bounds reflect that IDS is able to automatically exploit this structure.

The proofs of these bounds follow from our recent analysis of Thompson sampling, and the implied regret bounds are the same as those established for Thompson sampling. In particular, since $\Psi_t(\pi_t^{\text{IDS}}) \leqslant \Psi_t(\pi_t^{\text{TS}})$ where π^{TS} is the Thompson sampling policy, it is enough to bound $\Psi_t(\pi_t^{\text{TS}})$. Several bounds on the information ratio of TS were provided by Russo and Van Roy (2016), and we defer to that paper for the proofs. While the analysis is similar in the cases considered here, IDS outperforms Thompson sampling in simulation, and, as we highlighted in the previous section, is sometimes provably much more informationally efficient.

In addition to the bounds stated here, recent work by Bubeck et al. (2015) and Bubeck and Eldan (2016) bounds the information ratio when the reward function is convex, and uses this to study the order of regret in adversarial bandit convex optimization. This points to a broader potential of using information-ratio analysis to study the information complexity of general online optimization problems.

To simplify the exposition, our results are stated under the assumption that rewards are uniformly bounded. This effectively controls the worst-case variance of the reward distribution, and as shown in the appendix of Russo and Van Roy (2016), our results can be extended to the case where reward distributions are sub-Gaussian.

Assumption 1. $\sup_{\bar{y}\in\mathcal{Y}} R(\bar{y}) - \inf_{\underline{y}\in\mathcal{Y}} R(\underline{y}) \leqslant 1$.

5.2.1. Worst-Case Bound.

The next proposition shows that $\Psi_t(\pi_t^{\text{IDS}})$ is never larger than $|\mathcal{A}|/2$. That is, there is always an action sampling distribution $\pi \in \mathcal{D}(\mathcal{A})$ such that $\Delta_t(\pi)^2 \leqslant (|\mathcal{A}|/2)g_t(\pi)$. As we will show in the coming sections, the ratio between regret and information gain can be much smaller under specific information structures.

Proposition 2. *For any $t \in \mathbb{N}$, $\Psi_t(\pi_t^{\text{IDS}}) \leqslant |\mathcal{A}|/2$ almost surely.*

Combining Proposition 2 with Corollary 1 shows that $\mathbb{E}[\text{Regret}(T, \pi^{\text{IDS}})] \leqslant \sqrt{\frac{1}{2}|\mathcal{A}|H(\alpha_1)T}$.

5.2.2. Full Information.

Our focus in this paper is on problems with *partial feedback*. For such problems, what the decision maker observes depends on the actions selected, which leads to a tension between exploration and exploitation. Problems with full information arise as an extreme point of our formulation where the outcome $Y_{t,a}$ is perfectly revealed by observing $Y_{t,\tilde{a}}$ for some $\tilde{a} \neq a$; what is learned does not depend on the selected action. The next proposition shows that under full information, the minimal information ratio is bounded by 1/2.

Proposition 3. *Suppose for each $t \in \mathbb{N}$ there is a random variable $Z_t \colon \Omega \to \mathcal{Z}$ such that for each $a \in \mathcal{A}$, $Y_{t,a} = (a, Z_t)$. Then for all $t \in \mathbb{N}$, $\Psi_t(\pi_t^{\text{IDS}}) \leqslant \frac{1}{2}$ almost surely.*

Combining this result with Corollary 1 shows that $\mathbb{E}[\text{Regret}(T, \pi^{\text{IDS}})] \leqslant \sqrt{\frac{1}{2}H(\alpha_1)T}$. Further, a worst-case bound on the entropy of α_1 gives the inequality $\mathbb{E}[\text{Regret}(T, \pi^{\text{IDS}})] \leqslant \sqrt{\frac{1}{2}\log(|\mathcal{A}|)T}$. Dani et al. (2007) show this bound is order optimal, in the sense that for any time horizon T and number of actions $|\mathcal{A}|$ there exists a prior distribution over θ under which $\inf_\pi \mathbb{E}[\text{Regret}(T, \pi)] \geqslant c_0\sqrt{\log(|\mathcal{A}|)T}$ where c_0 is a numerical constant that does not depend on $|\mathcal{A}|$ or T. The bound here improves upon this worst-case bound since $H(\alpha_1)$ can be much smaller than $\log(|\mathcal{A}|)$ when the prior distribution is informative.

5.2.3. Linear Optimization Under Bandit Feedback.

The stochastic linear bandit problem has been widely studied (e.g., Dani et al. 2008, Rusmevichientong and Tsitsiklis 2010, Abbasi-Yadkori et al. 2011) and is one of the most important examples of a multi-armed bandit problem with "correlated arms." In this setting, each action is associated with a finite dimensional feature vector, and the mean reward generated by an action is the inner product between its known feature vector and some unknown parameter vector. Because of this structure, observations from taking one action allow the decision maker to make inferences about other actions. The next proposition bounds the minimal information ratio for such problems.

Proposition 4. *If $\mathcal{A} \subset \mathbb{R}^d$, $\Theta \subset \mathbb{R}^d$, and $\mathbb{E}[R_{t,a} \mid \theta] = a^T\theta$ for each action $a \in \mathcal{A}$, then $\Psi_t(\pi_t^{\text{IDS}}) \leqslant d/2$ almost surely for all $t \in \mathbb{N}$.*

This result shows the inequalities $\mathbb{E}[\text{Regret}(T, \pi^{\text{IDS}})] \leqslant \sqrt{\frac{1}{2}H(\alpha_1)dT} \leqslant \sqrt{\frac{1}{2}\log(|\mathcal{A}|)dT}$ for linear bandit problems. Dani et al. (2007) again show this bound is order optimal in the sense that, for any time horizon T and dimension d, when the action set is $\mathcal{A} = \{0,1\}^d$ there exists a prior distribution over θ such that $\inf_\pi \mathbb{E}[\text{Regret}(T,\pi)] \geqslant c_0\sqrt{\log(|\mathcal{A}|)dT}$ where c_0 is a constant that is independent of d and T. The bound here improves upon this worst-case bound since $H(\alpha_1)$ can be much smaller than $\log(|\mathcal{A}|)$ when the prior distribution in informative.

5.2.4. Combinatorial Action Sets and "Semi-Bandit" Feedback. To motivate the information structure studied here, consider a simple resource allocation problem. There are d possible projects, but the decision maker can allocate resources to at most $m \leqslant d$ of them at a time. At time t, project $i \in \{1,\dots,d\}$ yields a random reward $X_{t,i}$, and the reward from selecting a subset of projects $a \in \mathcal{A} \subset \{a' \subset \{0,1,\dots,d\}\colon |a'| \leqslant m\}$ is $m^{-1}\sum_{i\in\mathcal{A}} X_{t,i}$. In the linear bandit formulation of this problem, upon choosing a subset of projects a the agent would only observe the overall reward $m^{-1}\sum_{i\in a} X_{t,i}$. It may be natural instead to assume that the outcome of each selected project $(X_{t,i}\colon i \in a)$ is observed. This type of observation structure is sometimes called semi-bandit feedback (Audibert et al. 2014).

A naive application of Proposition 4 to address this problem would show $\Psi_t^* \leqslant d/2$. The next proposition shows that since the entire parameter vector $(\theta_{t,i}\colon i \in a)$ is observed upon selecting action a, we can provide an improved bound on the information ratio.

Proposition 5. *Suppose $\mathcal{A} \subset \{a \subset \{0,1,\dots,d\}\colon |a| \leqslant m\}$, and that there are random variables $(X_{t,i}\colon t \in \mathbb{N}, i \in \{1,\dots,d\})$ such that*

$$Y_{t,a} = (X_{t,i}\colon i \in a) \quad \text{and} \quad R_{t,a} = \frac{1}{m}\sum_{i\in a} X_{t,i}.$$

Assume that the random variables $\{X_{t,i}\colon i \in \{1,\dots,d\}\}$ are independent conditioned on $\mathcal{F}_t$ and $X_{t,i} \in [-1/2, 1/2]$ almost surely for each (t,i). Then for all $t \in \mathbb{N}$, $\Psi_t(\pi_t^{\text{IDS}}) \leqslant d/(2m^2)$ almost surely.

In this problem, there are as many as $\binom{d}{m}$ actions, but because IDS exploits the structure relating actions to one another, its regret is only polynomial in m and d. In particular, combining Proposition 5 with Corollary 1 shows $\mathbb{E}[\text{Regret}(T,\pi^{\text{IDS}})] \leqslant (1/m)\sqrt{(d/2)H(\alpha_1)T}$. Since $H(\alpha_1) \leqslant \log|\mathcal{A}| = O(m\log(d/m))$ this also yields a bound of order $\sqrt{(d/m)\log(d/m)T}$. As shown by Audibert et al. (2014), the lower bound[1] for this problem is of order $\sqrt{(d/m)T}$, so our bound is order optimal up to a $\sqrt{\log(d/m)}$ factor.

6. Computational Methods

IDS offers an abstract design principle that captures some key qualitative properties of the Bayes-optimal solution while accommodating tractable computation for many relevant problem classes. However, additional work is required to design efficient computational methods that implement IDS for specific problem classes. In this section, we provide guidance and examples.

We will focus in this section on the problem of generating an action A_t given the posterior distribution over θ at time t. This sidesteps the problem of computing and representing a posterior distribution, which can present its own challenges. Though IDS could be combined with approximate Bayesian inference methods, we will focus here on the simpler context in which posterior distributions can be efficiently computed and stored, as is the case when working with tractable finite uncertainty sets or appropriately chosen conjugate priors. It is worth noting, however, that two of our algorithms approximate IDS using samples from the posterior distribution, and this may be feasible through the use of Markov chain Monte Carlo even in cases where the posterior distribution cannot be computed or even stored.

6.1. Evaluating the Information Ratio

Given a finite action set $\mathcal{A} = \{1,\dots,K\}$, we can view an action distribution π as a K-dimensional vector of probabilities. The information ratio can then be written as

$$\Psi_t(\pi) = \frac{(\pi^\top\vec{\Delta})^2}{\pi^\top\vec{g}},$$

where $\vec{\Delta}$ and $\vec{g}$ are K-dimensional vectors with components $\vec{\Delta}_k = \Delta_t(k)$ and $\vec{g}_k = g_t(k)$ for $k \in \mathcal{A}$. In this subsection, we discuss the computation of $\vec{\Delta}$ and $\vec{g}$ for use in evaluation of the information ratio.

There is no general efficient procedure for computing $\vec{\Delta}$ and $\vec{g}$ given a posterior distribution, because that would require computing integrals over possibly high-dimensional spaces. Such computation can often be carried out efficiently by leveraging the functional form of the specific posterior distribution and often requires numerical integration. To illustrate the design of problem-specific computational procedures, we will present two simple examples in this subsection.

We begin with a conceptually simple model involving finite uncertainty sets.

Example 7 (Finite Sets). Consider a problem in which θ takes values in $\Theta = \{1,\dots,L\}$, the action set is $\mathcal{A} = \{1,\dots,K\}$, the observation set is $\mathcal{Y} = \{1,\dots,N\}$, and the reward function $R\colon \mathcal{Y} \mapsto \mathbb{R}$ is arbitrary. Let p_1 be the prior probability mass function of θ and let $q_{\theta,a}(y)$ be the probability, conditioned on θ, of observing y when action a is selected.

Note that the posterior probability mass function p_t, conditioned on observations made prior to period t, can be computed recursively via Bayes' rule:

$$p_{t+1}(\theta) \leftarrow \frac{p_t(\theta) q_{\theta, A_t}(Y_{t,A_t})}{\sum_{\theta' \in \Theta} p_t(\theta') q_{\theta', A_t}(Y_{t,A_t})}.$$

Given the posterior distribution p_t along with the model parameters (L, K, N, R, q), Algorithm 1 computes $\vec{\Delta}$ and $\vec{g}$. Line 1 computes the optimal action for each value of θ. Line 2 calculates the probability that each action is optimal. Line 3 computes the marginal distribution of $Y_{1,a}$ and line 4 computes the joint probability mass function of $(A^*, Y_{1,a})$. Lines 5 and 6 use the aforementioned probabilities to compute $\vec{\Delta}$ and $\vec{g}$.

Algorithm 1 (finiteIR(L, K, N, R, p, q))
1: $\Theta_a \leftarrow \{\theta \mid a = \arg\max_{a'} \sum_y q_{\theta,a'}(y) R(y)\}, \quad \forall\, \theta$
2: $p(a^*) \leftarrow \sum_{\theta \in \Theta_{a^*}} p(\theta), \quad \forall\, a^*$
3: $p_a(y) \leftarrow \sum_\theta p(\theta) q_{\theta,a}(y), \quad \forall\, a, y, \theta$
4: $p_a(a^*, y) \leftarrow \frac{1}{p(a^*)} \sum_{\theta \in \Theta_{a^*}} q_{\theta,a}(y), \quad \forall\, a, y, a^*$
5: $R^* \leftarrow \sum_a \sum_{\theta \in \Theta_a} \sum_y p(\theta) q_{\theta,a}(y) R(y)$
6: $\vec{g}_a \leftarrow \sum_{a^*, y} p_a(a^*, y) \log \frac{p_a(a^*, y)}{p(a^*) p_a(y)}, \quad \forall\, a$
7: $\vec{\Delta}_a \leftarrow R^* - \sum_\theta p(\theta) \sum_y q_{\theta,a}(y) R(y), \quad \forall\, a$
8: **return** $\vec{\Delta}, \vec{g}$.

Next, we consider the beta-Bernoulli bandit.

Example 8 (Beta-Bernoulli Bandit). Consider a multi-armed bandit problem with binary rewards: $\mathcal{A} = \{1, \ldots, K\}$, $\mathcal{Y} = \{0, 1\}$, and $R(y) = y$. Model parameters $\theta \in \mathbb{R}^K$ specify the mean reward θ_a of each action a. Components of θ are independent and each beta-distributed with prior parameters $\beta_1^1, \beta_1^2 \in \mathbb{R}_+^K$.

Because the beta distribution is a conjugate prior for the Bernoulli distribution, the posterior distribution of each θ_a is a beta distribution. The posterior parameters $\beta_{t,a}^1, \beta_{t,a}^2 \in \mathbb{R}_+$ can be computed recursively:

$$(\beta_{t+1,a}^1, \beta_{t+1,a}^2) \leftarrow \begin{cases} (\beta_{t,a}^1 + Y_{t,a}, \beta_{t,a}^2 + (1 - Y_{t,a})) & \text{if } A_t = a, \\ (\beta_{t,a}^1, \beta_{t,a}^2) & \text{otherwise.} \end{cases}$$

Given the posterior parameters (β_t^1, β_t^2), Algorithm 2 computes $\vec{\Delta}$ and $\vec{g}$.

Line 5 of the algorithm computes the posterior probability mass function of A^*. It is easy to derive the expression used:

$$\begin{aligned} \mathbb{P}_t(A^* = a) &= \mathbb{P}_t\left(\bigcap_{a' \neq a} \{\theta_{a'} \leqslant \theta_a\}\right) \\ &= \int_0^1 f_a(x) \mathbb{P}_t\left(\bigcap_{a' \neq a} \{\theta_{a'} \leqslant x\} \,\middle|\, \theta_a = x\right) dx \\ &= \int_0^1 f_a(x) \left(\prod_{a' \neq a} F_{a'}(x)\right) dx \\ &= \int_0^1 \left[\frac{f_a(x)}{F_a(x)}\right] \bar{F}(x)\, dx, \end{aligned}$$

where f_a, F_a, and $\bar{F}$ are defined as in lines 1–3 of the algorithm, with arguments $(K, \beta_t^1, \beta_t^2)$. Using expressions that can be derived in a similar manner, for each pair of actions Lines 6–7 compute $M_{a' \mid a} := \mathbb{E}_t[\theta_{a'} \mid \theta_a = \max_{a''} \theta_{a''}]$, the expected value of $\theta_{a'}$ given that action a is optimal. Lines 8–9 compute the expected reward of the optimal action $\rho^* = \mathbb{E}_t[\max_a \theta_a]$ and uses that to compute, for each action,

$$\vec{\Delta}_a = \mathbb{E}_t\left[\max_{a'} \theta_a - \theta_a\right] = \rho^* - \frac{\beta_{t,a}^1}{(\beta_{t,a}^1 + \beta_{t,a}^2)}.$$

Finally, line 10 computes $\vec{g}$. The expression makes use of the following fact, which is a consequence of standard properties of mutual information:[2]

$$I_t(A^*; Y_{t,a}) = \sum_{a^* \in \mathcal{A}} \mathbb{P}_t(A^* = a^*) \cdot D_{\mathrm{KL}}\left(\mathbb{P}_t(Y_{t,a} = \cdot \mid A^* = a^*) \,\|\, \mathbb{P}_t(Y_{t,a} = \cdot)\right). \quad (7)$$

That is, the mutual information between A^* and $Y_{t,a}$ is the expected Kullback-Leibler divergence between the posterior predictive distribution $\mathbb{P}_t(Y_{t,a} = \cdot)$ and the predictive distribution conditioned on the identity of the optimal action $\mathbb{P}_t(Y_{t,a} = \cdot \mid A^* = a^*)$. For our beta-Bernoulli model, the information gain $\vec{g}_a$ is the expected Kullback-Leibler divergence between a Bernoulli distribution with mean $M_{a \mid A^*}$ and the posterior distribution at action a, which is Bernoulli with parameter $\beta_{t,a}^1 / (\beta_{t,a}^1 + \beta_{t,a}^2)$.

Algorithm 2, as we have presented it, is somewhat abstract and can not readily be implemented on a computer. In particular, lines 1–4 require computing and storing functions of a continuous variable and several lines require integration of continuous functions. However, near-exact approximations can be efficiently generated by evaluating integrands at discrete grid of points $\{x^1, \ldots, x^n\} \subset [0, 1]$. The values of $f_a(x), F_a(x), G_a(x)$, and $\bar{F}(x)$ can be computed and stored for each value in this grid. The compute time can also be reduced via memoization, since values change only for one action per time period. The compute time of such an implementation scales with $K^2 n$ where K is the number of actions and n is the number of points used in the discretization of $[0, 1]$. The bottleneck is Line 7.

Algorithm 2 (betaBernoulliIR(K, β^1, β^2))
1: $f_a(x) \leftarrow \text{beta.pdf}(x \mid \beta_a^1, \beta_a^2), \quad \forall\, a, x$
2: $F_a(x) \leftarrow \text{beta.cdf}(x \mid \beta_a^1, \beta_a^2), \quad \forall\, a, x$
3: $\bar{F}(x) \leftarrow \prod_a F_a(x), \quad \forall\, x$
4: $G_a(x) \leftarrow \int_0^x y f_a(y)\, dy, \quad \forall\, a, x$

5: $p^*(a) \leftarrow \int_0^1 \left[\frac{f_a(x)}{F_a(x)}\right] \bar{F}(x)\,dx, \quad \forall\, a$

6: $M_{a\,|\,a} \leftarrow \frac{1}{p^*(a)} \int_0^1 \left[\frac{x f_a(x)}{F_a(x)}\right] \bar{F}(x)\,dx, \quad \forall\, a$

7: $M_{a'\,|\,a} \leftarrow \frac{1}{p^*(a)} \int_0^1 \left[\frac{f_a(x)\bar{F}(x)}{F_a(x)F_{a'}(x)}\right] G_{a'}(x)\,dx m$
$\forall\, a, a' \neq a$

8: $\rho^* \leftarrow \sum_a p^*(a) M_{a\,|\,a}$

9: $\vec{\Delta}_a \leftarrow \rho^* - \frac{\beta_a^1}{\beta_a^1 + \beta_a^2}, \quad \forall\, a$

10: $\vec{g}_a \leftarrow \sum_{a'} p^*(a')(M_{a\,|\,a'} \log(M_{a\,|\,a'}(\beta_a^1 + \beta_a^2)/\beta_a^1)$
$+ (1 - M_{a\,|\,a'}) \log((1 - M_{a\,|\,a'})(\beta_a^1 + \beta_a^2)/\beta_a^2)), \quad \forall\, a$

11: **return** $\vec{\Delta}, \vec{g}$.

6.2. Optimizing the Information Ratio

Let us now discuss how to generate an action given $\vec{\Delta}$ and $\vec{g} \neq 0$. If $\vec{g} = 0$, the optimal action is known with certainty, and therefore the action selection problem is trivial. Otherwise, IDS selects an action by solving

$$\min_{\pi \in \mathcal{S}_K} \frac{(\pi^\top \vec{\Delta})^2}{\pi^\top \vec{g}}, \tag{8}$$

where $\mathcal{S}_K = \{\pi \in \mathbb{R}_+^K : \sum_k \pi_k = 1\}$ is the K-dimensional unit simplex, and samples from the resulting distribution π.

The following result establishes that (8) is a convex optimization problem and, surprisingly, has an optimal solution with at most two nonzero components. Therefore, while IDS is a randomized policy, it suffices to randomize over two actions.

Proposition 6. *For all $\vec{\Delta}, \vec{g} \in \mathbb{R}_+^K$ such that $\vec{g} \neq 0$, the function $\pi \mapsto (\pi^\top \vec{\Delta})^2 / \pi^\top \vec{g}$ is convex on $\{\pi \in \mathbb{R}^K : \pi^\top \vec{g} > 0\}$. Moreover, this function is minimized over $\mathcal{S}_K$ by some π^* for which $|\{k : \pi_k^* > 0\}| \leqslant 2$.*

Algorithm 3 leverages Proposition 6 to efficiently choose an action in a manner that minimizes (6). The algorithm takes as input $\vec{\Delta} \in \mathbb{R}_+^K$ and $\vec{g} \in \mathbb{R}_+^K$, which provide the expected regret and information gain of each action. The sampling distribution that minimizes (6) is computed by iterating over all pairs of actions $(a, a') \in \mathcal{A} \times \mathcal{A}$, and for each, computing the probability q that minimizes the information ratio among distributions that sample a with probability q and a' with probability $1 - q$. This one-dimensional optimization problem requires little computation since the objective is convex; q can be computed by solving for the first-order necessary condition or approximated by a bisection method. The compute time of this algorithm scales with K^2.

Algorithm 3 (IDSAction($K, \vec{\Delta}, \vec{g}$))

1: $q_{a,a'} \leftarrow \arg\min_{q' \in [0,1]} [q'\vec{\Delta}_a + (1 - q')\vec{\Delta}_{a'}]^2 /$
$[q'\vec{g}_a + (1 - q')\vec{g}_{a'}], \quad \forall\, a < K, a' > a$

2: $(a^*, a^{**}) \leftarrow \arg\min_{a<K, a'>a} [q_{a,a'}\vec{\Delta}_a + (1 - q_{a,a'})\vec{\Delta}_{a'}]^2 /$
$[q_{a,a'}\vec{g}_a + (1 - q_{a,a'})\vec{g}_{a'}]$

3: Sample $b \sim$ Bernoulli$(q_{a^*,a^{**}})$

4: **return** $ba^* + (1 - b)a^{**}$.

6.3. Approximating the Information Ratio

Though reasonably efficient algorithms can be devised to implement IDS for various problem classes, some applications, such as those arising in high-throughput web services, call for extremely fast computation. As such, it is worth considering approximations to the information ratio that retain salient features while enabling faster computation. In this section, we discuss some useful approximation concepts.

The dominant source of complexity in computing $\vec{\Delta}$ and $\vec{g}$ is in the calculation of requisite integrals, which can require integration over high-dimensional spaces. One approach to addressing this challenge is to replace integrals with sample-based estimates. Algorithm 4 does this. In addition to the number of actions K and routines for evaluation q and R, the algorithm takes as input M representative samples of θ. In the simplest use scenario, these would be independent samples drawn from the posterior distribution. The steps correspond to those of Algorithm 1, but with the set of possible models approximated by the set of representative samples. For many problems, even when exact computation of $\vec{\Delta}$ and $\vec{g}$ is intractable due to required integration over high-dimensional spaces, Algorithm 1 can generate close approximations from a moderate number of samples M.

Algorithm 4 (SampleIR($K, q, R, M, \theta^1, \ldots, \theta^M$))

1: $\hat{\Theta}_a \leftarrow \{m \,|\, a = \arg\max_{a'} \sum_y q_{\theta^m, a'}(y) R(y)\}$

2: $\hat{p}(a^*) \leftarrow |\hat{\Theta}_{a^*}|/M, \quad \forall\, a^*$

3: $\hat{p}_a(y) \leftarrow \sum_m q_{a,\theta^m}(y)/M, \quad \forall\, y$

4: $\hat{p}_a(a^*, y) \leftarrow \sum_{m \in \Theta_a} q_{a,\theta^m}(y)/M, \quad \forall\, a^*, y$

5: $\hat{R}^* \leftarrow \sum_{a,y} \hat{p}_a(a, y) R(y)$

6: $\vec{g}_a \leftarrow \sum_{a^*,y} \hat{p}_a(a^*, y) \log \frac{\hat{p}_a(a^*, y)}{\hat{p}(a^*)\hat{p}_a(y)}, \quad \forall\, a$

7: $\vec{\Delta}_a \leftarrow R^* - M^{-1} \sum_m \sum_y q_{\theta^m, a}(y) R(y), \quad \forall\, a$

8: **return** $\vec{\Delta}, \vec{g}$.

The information ratio is designed to effectively address indirect information, cumulating information, and irrelevant information, for a very broad class of learning problems. It can sometimes be helpful to replace the information ratio with alternative information measures that adequately address these issues for more specialized classes of problems. As an example, we will introduce the variance-based information ratio, which is suitable for some problems with bandit feedback, satisfies our regret bounds for such problems, and can facilitate design of more efficient numerical methods.

To motivate the variance-based information ratio, note that when rewards our bounded, with $R(y) \in [0,1]$ for all y, our information measure term is lower-bounded according to

$$\begin{aligned} g_t(a) &= I_t(A^*; Y_{t,a}) \\ &= \sum_{a^* \in \mathcal{A}} \mathbb{P}_t(A^* = a^*) \\ &\quad \cdot D_{\mathrm{KL}}(\mathbb{P}_t(Y_{t,a} = \cdot \mid A^* = a^*) \,||\, \mathbb{P}_t(Y_{t,a} = \cdot)) \\ &\geqslant \sum_{a^* \in \mathcal{A}} \mathbb{P}_t(A^* = a^*) \\ &\quad \cdot D_{\mathrm{KL}}(\mathbb{P}_t(R_{t,a} = \cdot \mid A^* = a^*) \,||\, \mathbb{P}_t(R_{t,a} = \cdot)) \\ &\overset{(a)}{\geqslant} 2 \sum_{a^* \in \mathcal{A}} \mathbb{P}_t(A^* = a^*)(\mathbb{E}_t[R_{t,a} \mid A^* = a^*] - \mathbb{E}_t[R_{t,a}])^2 \\ &= 2\mathbb{E}_t[(\mathbb{E}_t[R_{t,a} \mid A^*] - \mathbb{E}_t[R_{t,a}])^2] \\ &= 2\mathrm{Var}_t(\mathbb{E}_t[R_{t,a} \mid A^*]), \end{aligned}$$

where $\mathrm{Var}_t(X) = \mathbb{E}_t[(X - \mathbb{E}_t[X])^2]$ denotes the variance of X under the posterior distribution. Inequality (a) is a simple corollary of Pinsker's inequality, and is given in Russo and Van Roy (2016, Fact 9). Let $v_t(a) := \mathrm{Var}_t(\mathbb{E}_t[R_{t,a} \mid A^*])$, which represents the variance of the conditional expectation $\mathbb{E}_t[R_{t,a} \mid A^*]$ under the posterior distribution. This measures how much the expected reward generated by action a varies depending on the identity of the optimal action A^*. The above lower bound on mutual information indicates that actions with high variance $v_t(a)$ must yield substantial information about which action is optimal. It is natural to consider an approximation to IDS that uses a variance-based information ratio:

$$\min_{\pi \in \mathcal{S}_K} \frac{(\pi^\top \vec{\Delta})^2}{\pi^\top \vec{v}},$$

where $\vec{v}_a = v_t(a)$.

While variance-based IDS will not minimize the information ratio, the next proposition establishes that it satisfies the bounds on the information ratio given by Propositions 2 and 4.

Proposition 7. *Suppose* $\sup_y R(y) - \inf_y R(y) \leqslant 1$ *and*

$$\pi_t \in \arg\min_{\pi \in \mathcal{S}_K} \frac{\Delta_t(\pi)^2}{v_t(\pi)}.$$

Then $\Psi_t(\pi_t) \leqslant |\mathcal{A}|/2$. *Moreover, if* $\mathcal{A} \subset \mathbb{R}^d$, $\Theta \subset \mathbb{R}^d$, *and* $\mathbb{E}[R_{t,a} | \theta] = a^T\theta$ *for each action* $a \in \mathcal{A}$, *then* $\Psi_t(\pi_t) \leqslant d/2$.

We now consider a couple of examples that illustrate computation of $\vec{v}$ and benefits of using this approximation. Our first example is the independent Gaussian bandit problem.

Example 9 (Independent Gaussian Bandit). Consider a multi-armed bandit problem with $\mathcal{A} = \{1, \ldots, K\}$, $\mathcal{Y} = \mathbb{R}$, and $R(y) = y$. Model parameters $\theta \in \mathbb{R}^K$ specify the mean reward θ_a of each action a. Components of θ are independent and Gaussian distributed, with prior means $\mu_1 \in \mathbb{R}^K$ and covariances $\sigma_1^2 \in \mathbb{R}^K$. When an action A_t is applied, the observation Y_t is drawn independently from $N(\theta_{A_t}, \eta^2)$.

The posterior distribution of θ is Gaussian, with independent components. Parameters can be computed recursively according to

$$\mu_{t+1,a} \leftarrow \begin{cases} \left(\frac{\mu_{t,a}}{\sigma_{t,a}^2} + \frac{Y_{t,a}}{\eta^2}\right) \Big/ \left(\frac{1}{\sigma_{t,a}^2} + \frac{1}{\eta^2}\right) & \text{if } A_t = a, \\ \mu_{t,a} & \text{otherwise,} \end{cases}$$

$$\sigma_{t+1,a} \leftarrow \begin{cases} \left(\frac{1}{\sigma_{t,a}^2} + \frac{1}{\eta^2}\right)^{-1} & \text{if } A_t = a, \\ \sigma_{t,a} & \text{otherwise.} \end{cases}$$

Given arguments (K, μ_t, σ_t), Algorithm 5 computes $\vec{\Delta}$ and $\vec{v}$ for the independent Gaussian bandit problem. Note that this algorithm is very similar to Algorithm 2, which was designed for the beta-Bernoulli bandit. One difference is that Algorithm 5 computes the variance-based information measure. In addition, the Gaussian distribution exhibits a special structure that simplifies the computation of $M_{a'|a} := \mathbb{E}_t[\theta_{a'} \mid \theta_a = \max_{a''} \theta_{a''}]$. In particular, the computation of $M_{a'|a}$ uses the following closed-form expression for the expected value of a truncated Gaussian distribution with mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$:

$$\begin{aligned} \mathbb{E}[X \mid X \leqslant x] &= \tilde{\mu} - \tilde{\sigma}\phi\left(\frac{x - \tilde{\mu}}{\tilde{\sigma}}\right) \Big/ \Phi\left(\frac{x - \tilde{\mu}}{\tilde{\sigma}}\right) \\ &= \tilde{\mu} - \tilde{\sigma}^2 f(x)/F(x), \end{aligned}$$

where $X \sim N(\tilde{\mu}, \tilde{\sigma}^2)$ and f and F are the probability density and cumulative distribution functions.

The analogous calculation that would be required to compute the standard information ratio is more complex.

Algorithm 5 (independentGaussianVIR(K, μ, σ))

1: $f_a(x) \leftarrow \text{Gaussian.pdf}(x \mid \mu_a, \sigma_a^2), \quad \forall\, a, x$
2: $F_a(x) \leftarrow \text{Gaussian.cdf}(x \mid \mu_a, \sigma_a^2), \quad \forall\, a, x$
3: $\bar{F}(x) \leftarrow \prod_a F_a(x), \quad \forall\, x$
4: $p^*(a) \leftarrow \int_0^1 \left[\frac{f_a(x)}{F_a(x)}\right] \bar{F}(x)\, dx, \quad \forall\, a$
5: $M_{a|a} \leftarrow \frac{1}{p^*(a)} \int_{-\infty}^{\infty} \left[\frac{x f_a(x)}{F_a(x)}\right] \bar{F}(x)\, dx, \quad \forall\, a$
6: $M_{a'|a} \leftarrow \mu_{a'} - \frac{\sigma_{a'}^2}{p^*(a)} \int_{-\infty}^{\infty} \left[\frac{f_a(x) f_{a'}(x)}{F_a(x) F_{a'}(x)}\right] \bar{F}(x)\, dx, \quad \forall\, a, a' \neq a$
7: $\rho^* \leftarrow \sum_a p^*(a) M_{a|a}$
8: $\Delta_a \leftarrow \rho^* - \mu_a, \quad \forall\, a$
9: $v_a \leftarrow \sum_{a'} p^*(a')(M_{a|a'} - \mu_a)^2, \quad \forall\, a$
10: **return** $\vec{\Delta}, \vec{v}$.

We next consider the linear bandit problem.

Example 10 (Linear Bandit). Consider a multi-armed bandit problem with $\mathcal{A} = \{1, \ldots, K\}$, $\mathcal{Y} = \mathbb{R}$, and $R(y) = y$. Model parameters $\theta \in \mathbb{R}^K$ are drawn from a Gaussian prior with mean μ_1 and covariance matrix Σ_1. There is a known matrix $\Phi = [\Phi_1, \cdots, \Phi_K] \in \mathbb{R}^{d \times K}$ such that, when an action A_t is applied, the observation Y_{t,A_t} is drawn independently from $N(\Phi_{A_t}\theta, \eta^2)$, where Φ_{A_t} denotes the A_tth column of Φ.

The posterior distribution of θ is Gaussian and can be computed recursively:

$$\mu_{t+1} = (\Sigma_t^{-1} + \Phi_{A_t}\Phi_{A_t}^\top/\eta^2)^{-1}(\Sigma_t^{-1}\mu_t + Y_{t,A_t}\Phi_{A_t}/\eta^2),$$
$$\Sigma_{t+1} = (\Sigma_t^{-1} + \Phi_{A_t}\Phi_{A_t}^\top/\eta^2)^{-1}.$$

We will develop an algorithm that leverages the fact that, for the linear bandit, $v_t(a)$ takes on a particularly simple form:

$$\begin{aligned} v_t(a) &= \mathrm{Var}_t(\mathbb{E}_t[R_{t,a} | A^*]) \\ &= \mathrm{Var}_t(\mathbb{E}_t[\Phi_a^\top \theta \mid A^*]) \\ &= \mathrm{Var}_t(\Phi_a^\top \mathbb{E}_t[\theta \mid A^*]) \\ &= \Phi_a^\top \mathbb{E}_t[(\mu_t^{A^*} - \mu_t)(\mu_t^{A^*} - \mu_t)^\top]\Phi_a \\ &= \Phi_a^\top L_t \Phi_a, \end{aligned}$$

where $\mu_t^a = \mathbb{E}_t[\theta | A^* = a]$ and $L_t = \mathbb{E}_t[(\mu_t^{A^*} - \mu_t)(\mu_t^{A^*} - \mu_t)^\top]$. Algorithm 6 presents a sample-based approach to computing $\vec{\Delta}$ and $\vec{v}$. In addition to model dimensions K and d and the problem data matrix Φ, the algorithm takes as input M representative values of θ, which in the simplest use scenario would be independent samples drawn from the posterior distribution $N(\mu_t, \Sigma_t)$. The algorithm approximates posterior means μ_t and μ_t^a as well as L_t by averaging suitable expressions over these samples. Due to the quadratic structure of $v_t(a)$, these calculations are substantially simpler than those that would be carried out by Algorithm 4, specialized to this context.

Algorithm 6 (linearSampleVIR$(K, d, M, \theta^1, \ldots, \theta^M)$)

$\hat{\mu} \leftarrow \sum_m \theta^m / M$
$\hat{\Theta}_a \leftarrow \{m : (\Phi^\top \theta^m)_a = \max_{a'} (\Phi\theta^m)_{a'}\}, \quad \forall a$
$\hat{p}^*(a) \leftarrow |\hat{\Theta}_a|/M, \quad \forall a$
$\hat{\mu}^a \leftarrow \sum_{\theta \in \hat{\Theta}_a} \theta / |\hat{\Theta}_a|, \quad \forall a$
$\hat{L} \leftarrow \sum_a \hat{p}^*(a)(\hat{\mu}^a - \hat{\mu})(\hat{\mu}^a - \hat{\mu})^\top$
$\rho^* \leftarrow \sum_a \hat{p}^*(a)\Phi_a^\top \hat{\mu}^a$
$\vec{v}_a \leftarrow \Phi_a^\top \hat{L} \Phi_a^\top, \quad \forall a$
$\vec{\Delta}_a \leftarrow \rho^* - \Phi_a^\top \hat{\mu}, \quad \forall a$
return $\vec{\Delta}, \vec{v}$.

It is interesting to note that Algorithms 4 and 6 do not rely on any special structure in the posterior distribution. Indeed, these algorithms should prove effective regardless of the form taken by the posterior. This points to a broader opportunity to use IDS or approximations to address complex models for which posteriors can not be efficiently computed or even stored, but for which it is possible to generate posterior samples via Markov chain Monte Carlo methods. We leave this as a future research opportunity.

7. Computational Results

This section presents computational results from experiments that evaluate the effectiveness of information-directed sampling in comparison to alternative algorithms. In Section 4.3, we showed that alternative approaches like UCB algorithms, Thompson sampling, and the knowledge gradient algorithm can perform very poorly when faced with complicated information structures and for this reason can be dramatically outperformed by IDS. In this section, we focus instead on simpler settings where current approaches are extremely effective. We find that even for these simple and widely studied settings, information-directed sampling displays state-of-the-art performance. For each experiment, the algorithm used to implement IDS is presented in the previous section.

IDS, TS, and some UCB algorithms, do not take the horizon T as input, and are instead designed to work well for all sufficiently long horizons. Other algorithms we simulate were optimized for the particular horizon of the simulation trial. The KG and KG* algorithms in particular, treat the simulation horizon as known, and explore less aggressively in later periods. We have tried to clearly delineate which algorithms are optimized for simulation horizon. We believe one can also design variants of IDS, TS, and UCB algorithms that reduce exploration as the time remaining diminishes, but we leave this for future work.

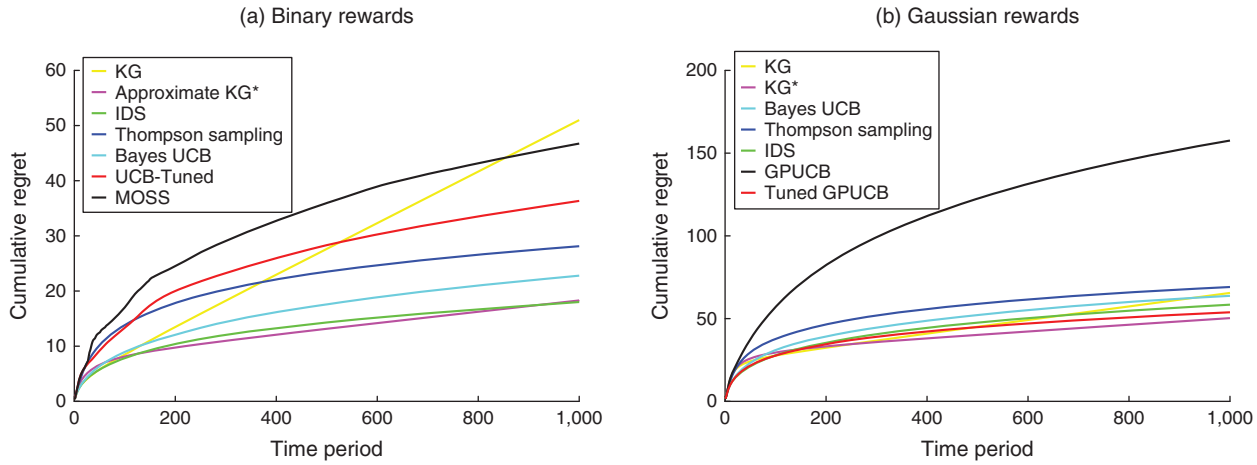
7.1. Beta-Bernoulli Bandit

Our first experiment involves a multi-armed bandit problem with independent arms and binary rewards. The mean reward of each arm is drawn from Beta(1, 1), which is the uniform distribution, and the means of separate arms are independent. Figure 1(a) and Table 1 present the results of 1,000 independent trials of an experiment with 10 arms and a time horizon of 1,000. We compared the performance of IDS to that of six other algorithms and found that it had the lowest average regret of 18.0.

The UCB1 algorithm of Auer et al. (2002) selects the action a, which maximizes the upper confidence bound $\hat{\theta}_t(a) + \sqrt{2\log(t)/N_t(a)}$ where $\hat{\theta}_t(a)$ is the empirical average reward from samples of action a and $N_t(a)$ is the number of samples of action a up to time t. The average regret of this algorithm is 130.7, which is dramatically larger than that of IDS. For this reason UCB1 is omitted from Figure 1(a).

The confidence bounds of UCB1 are constructed to facilitate theoretical analysis. For practical performance Auer et al. (2002) proposed using an algorithm

Figure 1. (Color online) Average Cumulative Regret Over 1,000 Trials



called UCB-Tuned. This algorithm selects the action a, which maximizes the upper confidence bound $\hat{\theta}_t(a) + \sqrt{\min\{1/4, \bar{V}_t(a)\}\log(t)/N_t(a)}$, where $\bar{V}_t(a)$ is an upper bound on the variance of the reward distribution at action a. While this method dramatically outperforms UCB1, it is still outperformed by IDS. The MOSS algorithm of Audibert and Bubeck (2009) is similar to UCB1 and UCB–Tuned, but uses slightly different confidence bounds. It is known to satisfy regret bounds for this problem that are minimax optimal up to a numerical constant factor.

In previous numerical experiments (Scott 2010; Kaufmann et al. 2012b, a; Chapelle and Li 2011), Thompson sampling and Bayes UCB exhibited state-of-the-art performance for this problem. Each also satisfies strong theoretical guarantees, and is known to be asymptotically optimal in the sense defined by Lai and Robbins (1985). Unsurprisingly, they are the closest competitors to IDS. The Bayes UCB algorithm, studied in Kaufmann et al. (2012a), constructs upper confidence bounds based on the quantiles of the posterior distribution: at time step t the upper confidence bound at an action is the $1 - 1/t$ quantile of the posterior distribution of that action.[3]

A somewhat different approach is the KG policy of Powell and Ryzhov (2012), which uses a one-step look-ahead approximation to the value of information to guide experimentation. For reasons described in Section 4.3.3, KG does not explore sufficiently to identify the optimal arm in this problem, and therefore its regret grows linearly with time. Because KG explores very little, its realized regret is highly variable, as depicted in Table 1. In 200 out of the 2,000 trials, the regret of KG was lower than 0.7, reflecting that the best arm was almost always chosen. In the worst 200 out of the 2,000 trials, the regret of KG was larger than 159.

KG is particularly poorly suited to problems with discrete observations and long time horizons. The KG* heuristic of Ryzhov et al. (2010) offers much better performance in some of these problems. At time t, KG* calculates the value of sampling an arm for $M \in \{1, \ldots, T - t\}$ periods and choosing the arm with the highest posterior mean in subsequent periods. It selects an action by maximizing this quantity over all possible arms and possible exploration lengths M. Our simulations require computing $T = 1{,}000$ decisions per trial, and a direct implementation of KG* requires order T^3 basic operations per decision. To enable

Table 1. Realized Regret Over 2,000 Trials in Bernoulli Experiment

	Time horizon agnostic						Optimized for time horizon		
Algorithm	IDS	V-IDS	TS	Bayes UCB	UCB1	UCB-Tuned	MOSS	KG	KG*
Mean regret	18.0	18.1	28.1	22.8	130.7	36.3	46.7	51.0	18.4
Standard error	0.4	0.4	0.3	0.3	0.4	0.3	0.2	1.5	0.6
Quantile 0.10	3.6	5.2	13.6	8.5	104.2	24.0	36.2	0.7	2.9
Quantile 0.25	7.4	8.1	18.0	12.5	117.6	29.2	40.0	2.9	5.4
Quantile 0.50	13.3	13.5	25.3	20.1	131.6	35.2	45.2	11.9	8.7
Quantile 0.75	22.5	22.3	35.0	30.6	144.8	41.9	51.0	82.3	16.3
Quantile 0.90	35.6	36.5	46.4	40.5	154.9	49.5	57.9	159.0	46.9
Quantile 0.95	51.9	48.8	53.9	47.0	160.4	54.9	64.3	204.2	76.6

efficient simulation, we use a heuristic approach to computing KG* proposed by Kamiński (2015). The approximate KG* algorithm we implement uses golden section search to maximize a nonconcave function, but is still empirically effective.

Finally, as demonstrated in Figure 1(a), variance-based IDS offers performance very similar to standard IDS for this problem.

It is worth pointing out that, although Gittins' indices characterize the Bayes optimal policy for infinite horizon discounted problems, the finite horizon formulation considered here is computationally intractable (Gittins et al. 2011). A similar index policy (Niño-Mora 2011) designed for finite horizon problems could be applied as a heuristic in this setting. However, with long time horizons, the associated computational requirements become onerous.

7.2. Independent Gaussian Bandit

Our second experiment treats a different multi-armed bandit problem with independent arms. The reward value at each action a follows a Gaussian distribution $N(\theta_a, 1)$. The mean $\theta_a \sim N(0,1)$ is drawn from a Gaussian prior, and the means of different reward distributions are drawn independently. We ran 2,000 simulation trials of a problem with 10 arms. The results are displayed in Figure 1(b) and Table 2.

For this problem, we compare variance-based IDS against Thompson sampling, Bayes UCB, and KG. We use the variance-based variant of IDS because it affords us computational advantages.

We also simulated the GPUCB of Srinivas et al. (2012). This algorithm maximizes the upper confidence bound $\mu_t(a) + \sqrt{\beta_t}\sigma_t(a)$ where $\mu_t(a)$ and $\sigma_t(a)$ are the posterior mean and standard deviation of θ_a. They provide regret bounds that hold with probability at least $1-\delta$ when $\beta_t = 2\log(|\mathcal{A}|t^2\pi^2/6\delta)$. This value of β_t is far too large for practical performance, at least in this problem setting. The average regret of GPUCB[4] is 157.6, which is roughly almost three times that of V-IDS. For this reason, we considered a tuned version of GPUCB that sets $\beta_t = c\log(t)$. We ran 1,000 trials of many different values of c to find the value $c = 0.9$ with the lowest average regret for this problem. This tuned version of GPUCB had average regret of 53.8, which is slightly better than IDS.

The work on KG focuses almost entirely on problems with Gaussian reward distributions and Gaussian priors. We find KG performs better in this experiment than it did in the Bernoulli setting, and its average regret is competitive with that of IDS.

As in the Bernoulli setting, KG's realized regret is highly variable. The median regret of KG is the lowest of any algorithm, but in 100 of the 2,000 trials its regret exceeded 283—seemingly reflecting that the algorithm did not explore enough to identify the best action. The KG* heuristic explores more aggressively, and performs very well in this experiment.

KG is particularly effective over short time spans. Unlike information-directed sampling, KG takes the time horizon T as an input, and explores less aggressively when there are fewer time periods remaining. Table 3 compares the regret of KG and IDS over different time horizons. Even though IDS does not take the time horizon into account, it is competitive with KG, even over short horizons. We believe that IDS can be modified to exploit fixed and known time horizons more effectively, though we leave the matter for future research.

7.3. Asymptotic Optimality

The previous subsections present numerical examples in which IDS outperforms Bayes UCB and Thompson sampling for some problems with independent

Table 2. Realized Regret Over 2,000 Trials in Independent Gaussian Experiment

	Time horizon agnostic				Optimized for time horizon		
Algorithm	V-IDS	TS	Bayes UCB	GPUCB	Tuned GPUCB	KG	KG*
Mean regret	58.4	69.1	63.8	157.6	53.8	65.5	50.3
Standard error	1.7	0.8	0.7	0.9	1.4	2.9	1.9
Quantile 0.10	24.0	39.2	34.7	108.2	24.2	16.7	19.4
Quantile 0.25	30.3	47.6	43.2	130.0	30.1	20.8	24.0
Quantile 0.50	39.2	61.8	57.5	156.5	41.0	25.9	29.9
Quantile 0.75	56.3	80.6	76.5	184.2	58.9	36.4	40.3
Quantile 0.90	104.6	104.5	97.5	207.2	86.1	155.3	74.7
Quantile 0.95	158.1	126.5	116.7	222.7	112.2	283.9	155.6

Table 3. Competitive Performance Without Knowing the Time Horizon

Time horizon T	10	25	50	75	100	250	500	750	1,000	2,000
Regret of V-IDS	9.8	16.1	21.1	24.5	27.3	36.7	48.2	52.8	58.3	68.4
Regret of KG(T)	9.2	15.3	20.5	22.9	25.4	35.2	45.3	52.3	62.9	80.0

Note. Average cumulative regret over 2,000 trials in the independent Gaussian experiment.

arms. This is surprising since each of these algorithms is known, in a sense we will soon formalize, to be asymptotically optimal for these problems. This section presents simulation results over a much longer time horizon that suggest IDS scales in the same asymptotically optimal way.

We consider again a problem with binary rewards and independent actions. The action $a_i \in \{a_1, \dots, a_K\}$ yields in each time period a reward that is 1 with probability θ_i and 0 otherwise. The seminal work of Lai and Robbins (1985) provides the following asymptotic lower bound on regret of any policy π:

$$\liminf_{T\to\infty} \frac{\mathbb{E}[\text{Regret}(T,\pi)\,|\,\theta]}{\log T} \geqslant \sum_{a\neq A^*} \frac{\theta_{A^*} - \theta_a}{D_{\text{KL}}(\theta_{A^*}\,||\,\theta_a)} := c(\theta).$$

Note that we have conditioned on the parameter vector θ, indicating that this is a frequentist lower bound. Nevertheless, when applied with an independent uniform prior over θ, both Bayes UCB and Thompson sampling are known to attain this lower bound (Kaufmann et al. 2012a, b).

Our next numerical experiment fixes a problem with three actions and with $\theta = (0.3, 0.2, 0.1)$. We compare algorithms over 10,000 time periods. Because of the expense of running this experiment, we were only able to execute 200 independent trials. Each algorithm uses a uniform prior over θ. Our results, along with the asymptotic lower bound of $c(\theta)\log(T)$, are presented in Figure 2.

7.4. Linear Bandit Problems

Our final numerical experiment treats a linear bandit problem. Each action $a \in \mathbb{R}^5$ is defined by a five-dimensional feature vector. The reward of action a at time t is $a^T\theta + \epsilon_t$ where $\theta \sim N(0, 10I)$ is drawn from a multivariate Gaussian prior distribution, and $\epsilon_t \sim N(0,1)$ is independent Gaussian noise. In each period, only the reward of the selected action is observed. In our experiment, the action set $\mathcal{A}$ contains 30 actions, each with features drawn uniformly at random from $[-1/\sqrt{5}, 1/\sqrt{5}]$. The results displayed in Figure 3 and Table 4 compare regret across 2,000 independent trials.

We simulate variance-based IDS using the implementation presented in Algorithm 6. We compare its regret to six competing algorithms. Like IDS, GP-UCB and Thompson sampling satisfy strong regret bounds for this problem.[5] Both algorithms are significantly outperformed by IDS.

We also include Bayes UCB (Kaufmann et al. 2012a) and a version of GP-UCB that was tuned, as in Section 7.2, to minimize its average regret. Each of these displays performance that is competitive with that of IDS. These algorithms are heuristics, in the sense that the way their confidence bounds are constructed differ significantly from those of linear UCB algorithms that are known to satisfy theoretical guarantees.

As discussed in Section 7.2, unlike IDS, KG takes the time horizon T as an input, and explores less aggressively when there are fewer time periods remaining. Table 5 compares IDS to KG over several different time horizons. Even though IDS does not exploit knowledge of the time horizon, it is competitive with KG over short time horizons.

In this experiment, KG* appears to offer a small improvement over standard KG, but as shown in the next subsection, it is much more computationally burdensome. To save computational resources, we have only executed 500 independent trials of the KG* algorithm.

7.5. Runtime Comparison

We now compare the time required to compute decisions using the algorithms we have applied. In our experiments, Thompson sampling and UCB algorithms are extremely fast, sometimes requiring only a few microseconds to reach a decision. As expected, our implementation of IDS requires significantly more compute time. However, IDS often reaches a decision in only a small fraction of a second, which is tolerable in many application areas. In addition, IDS may be accelerated

Figure 2. (Color online) Cumulative Regret Over 200 Trials

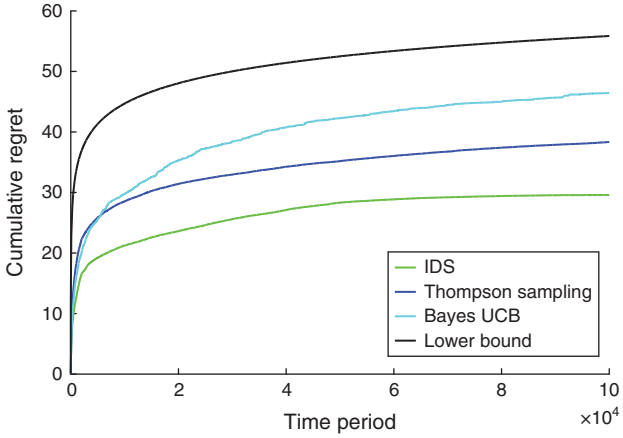


Figure 3. (Color online) Regret in Linear-Gaussian Model

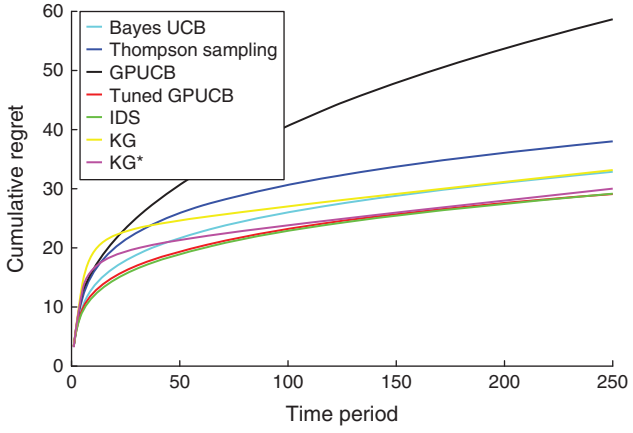


Table 4. Realized Regret Over 2,000 Trials in Linear Experiment

	Time horizon agnostic				Optimized for time horizon		
Algorithm	V-IDS	TS	Bayes UCB	GPUCB	Tuned GPUCB	KG	KG*
Mean regret	29.2	38.0	32.9	58.7	29.1	33.2	30.0
Standard error	0.5	0.4	0.4	0.3	0.4	0.7	1.4
Quantile 0.10	13.0	22.6	18.9	41.3	14.5	12.7	11.9
Quantile 0.25	17.6	27.6	23.1	48.9	18.4	17.5	16.1
Quantile 0.50	23.2	34.3	29.2	57.9	24.0	24.1	20.6
Quantile 0.75	32.1	43.7	39.0	67.4	32.9	34.5	28.5
Quantile 0.90	49.5	56.5	48.7	77.1	46.6	60.9	55.6
Quantile 0.95	67.5	67.5	58.4	82.7	59.9	94.5	96.1

Note. KG* results are over 500 trials.

Table 5. Competitive Performance Without Knowing the Time Horizon Average Cumulative Regret Over 2,000 Trials in Linear Gaussian Experiment

Time horizon T	10	25	50	75	100	250	500
Regret of V-IDS	11.8	16.2	19.6	21.6	23.3	31.1	34.7
Regret of KG(T)	11.1	15.1	19.0	22.5	24.1	34.4	43.0

considerably via parallel processing or an optimized implementation.

The results for KG are mixed. For independent Gaussian models, certain integrals can be computed via closed-form expressions, allowing KG to execute quickly. There is also a specialized numerical procedure for implementing KG for correlated (or linear) Gaussian models, but computation is an order of magnitude slower than in the independent case. For correlated Gaussian models, the KG* policy is much slower than both KG and IDS. For beta-Bernoulli problems, KG can be computed very easily, but yields poor performance. A direct implementation of the KG* policy was too slow to simulate, and so we have used a heuristic approach presented in (Kamiński 2015), which uses golden section search to maximize a function that is not necessarily unimodal. This method is labeled "Approx KG*" in Table 6.

Table 6 displays results for the Bernoulli experiment described in Section 7.1. It shows the average time required to compute a decision in a 1,000 period problem with 10, 30, 50, and 70 arms. IDS was implemented using Algorithm 2 to evaluate the information ratio, and Algorithm 3 to optimize it. The numerical integrals in Algorithm 2 were approximated using quadrature with 1,000 equally spaced points. Table 7 presents results of the corresponding experiment in the Gaussian case. Finally, Table 8 displays results for the linear bandit

Table 6. Bernoulli Experiment: Compute Time per Decision in Seconds

Arms	IDS	V-IDS	TS	Bayes UCB	UCB1	KG	Approx KG*
10	0.011013	0.01059	0.000025	0.000126	0.000008	0.000036	0.074618
30	0.047021	0.047529	0.000023	0.000147	0.000005	0.000017	0.215145
50	0.104328	0.10203	0.000024	0.000176	0.000005	0.000017	0.358505
70	0.18556	0.178689	0.000028	0.000167	0.000005	0.000017	0.494455

Table 7. Independent Gaussian Experiment: Compute Time per Decision in Seconds

Arms	V-IDS	TS	Bayes UCB	GPUCB	KG	KG*
10	0.00298	0.000008	0.00002	0.00001	0.000146	0.001188
30	0.012597	0.000005	0.000009	0.000005	0.000097	0.003157
50	0.023084	0.000006	0.000009	0.000005	0.000094	0.005146
70	0.03913	0.000006	0.000009	0.000005	0.000098	0.006364

Table 8. Linear Gaussian Experiment: Compute Time per Decision in Seconds

Arms	Dimension	V-IDS	TS	Bayes UCB	GPUCB	KG	KG*
15	3	0.004305	0.000178	0.000139	0.000048	0.002709	0.311935
30	5	0.008635	0.000064	0.000048	0.000038	0.004789	0.589998
50	20	0.026222	0.000077	0.000083	0.000068	0.008356	1.051552
100	30	0.079659	0.000115	0.000148	0.00013	0.017034	2.067123

experiments described in Section 7.4, which make use of Algorithm 6 and Markov chain Monte Carlo sampling with $M = 10{,}000$ samples. The table provides the average time required to compute a decision in a 250 period problem.

8. Conclusion

This paper has proposed information-directed sampling—a new algorithm for online optimization problems in which a decision maker must learn from partial feedback. We establish a general regret bound for the algorithm, and specialize this bound to several widely studied problem classes. We show that it sometimes greatly outperforms other popular approaches, which do not carefully measure the information provided by sampling actions. Finally, for some simple and widely studied classes of multi-armed bandit problems we demonstrate simulation performance surpassing popular approaches.

Many important open questions remain, however. IDS solves a single-period optimization problem as a proxy to an intractable multi-period problem. A solution of this single-period problem can itself be computationally demanding, especially in cases where the number of actions is enormous or mutual information is difficult to evaluate. An important direction for future research concerns the development of computationally elegant procedures to implement IDS in important cases. Even when the algorithm cannot be directly implemented, however, one may hope to develop simple algorithms that capture its main benefits. Proposition 1 shows that any algorithm with small information ratio satisfies strong regret bounds. Thompson sampling is a simple algorithm that, we conjecture, sometimes has nearly minimal information ratio. Perhaps simple schemes with small information ratio could be developed for other important problem classes, like the sparse linear bandit problem.

In addition to computational considerations, a number of statistical questions remain open. One question raised is whether IDS attains the lower bound of Lai and Robbins (1985) for some bandit problems with independent arms. Beyond the empirical evidence presented in Section 7.3, there are some theoretical reasons to conjecture this is true. Next, a more precise understanding of the problem's *information complexity* remains an important open question for the field. Our regret bound depends on the problem's information complexity through a term we call the information ratio, but it is unclear if or when this is the right measure. Finally, it may be possible to derive lower bounds using the same information-theoretic style of argument used in the derivation of our upper bounds.

9. Extensions

This section presents a number of ways in which the results and ideas discussed throughout this paper can be extended. We will consider the use of algorithms like information-directed sampling for pure-exploration problems, a form of information-directed sampling that aims to acquire information about θ instead of A^*, and a version of information-directed sampling that uses a tuning parameter to control how aggressively the algorithm explores. In each case, new theoretical guarantees can be easily established by leveraging our analysis of information-directed sampling.

9.1. Pure Exploration Problems

Consider the problem of adaptively gathering observations $(A_1, Y_{1,A_1}, \ldots, A_{T-1}, Y_{T-1,A_{T-1}})$ so as to minimize the expected loss of the best decision at time T,

$$\mathbb{E}\left[\min_{a\in\mathcal{A}} \Delta_T(a)\right]. \tag{9}$$

Recall that we have defined $\Delta_t(a) := \mathbb{E}[R_{t,A^*} - R_{t,a} \mid \mathcal{F}_t]$ to be the expected regret of action a at time t. This is a "pure exploration problem," in the sense that one is interested only in the terminal regret (9) and not in the algorithm's cumulative regret. However, the next proposition shows that bounds on the algorithm's cumulative expected regret imply bounds on $\mathbb{E}[\min_{a\in\mathcal{A}} \Delta_T(a)]$.

Proposition 8. *If actions are selected according to a policy π, then*

$$\mathbb{E}\left[\min_{a\in\mathcal{A}} \Delta_T(a)\right] \leqslant \frac{\mathbb{E}[\mathrm{Regret}(T,\pi)]}{T}.$$

Proof. By the tower property of conditional expectation, $\mathbb{E}[\Delta_{t+1}(a) \mid \mathcal{F}_t] = \Delta_t(a)$. Therefore, Jensen's inequality shows $\mathbb{E}[\min_{a\in\mathcal{A}} \Delta_{t+1}(a) \mid \mathcal{F}_t] \leqslant \min_{a\in\mathcal{A}} \Delta_t(a) \leqslant \Delta_t(\pi_t)$. Taking expectations and iterating this relation shows that

$$\mathbb{E}\left[\min_{a\in\mathcal{A}} \Delta_T(a)\right] \leqslant \mathbb{E}\left[\min_{a\in\mathcal{A}} \Delta_t(a)\right] \leqslant \mathbb{E}[\Delta_t(\pi_t)] \quad \forall\, t \in \{1,\ldots,T\}. \tag{10}$$

The result follows by summing both sides of (10) over $t \in \{1,\ldots,T\}$ and dividing each by T. □

Information-directed sampling is designed to have low cumulative regret, and therefore balances between acquiring information and taking actions with low expected regret. For pure exploration problems, it is natural instead to consider an algorithm that always acquires as much information about A^* as possible. The next proposition provides a theoretical guarantee for an algorithm of this form. The proof of this result combines our analysis of information-directed sampling with Proposition 8.

Proposition 9. *If actions are selected so that*

$$A_t \in \arg\max_{a\in\mathcal{A}} g_t(a),$$

and $\Psi_t^* \leqslant \lambda$ *almost surely for each* $t \in \{1,\dots,T\}$, *then*

$$\mathbb{E}\left[\min_{a\in\mathcal{A}} \Delta_T(a)\right] \leqslant \sqrt{\frac{\lambda H(\alpha_1)}{T}}.$$

Proof. To simplify notation, let $\Delta_t^* = \min_{a\in\mathcal{A}} \Delta_t(a)$ denote the minimal expected regret at time t, and $g_t^* = \max_{a\in\mathcal{A}} g_t(a)$ denote the information gain under the current algorithm.

Since $\Delta_t(\pi_t^{\text{IDS}})^2 \leqslant \lambda g_t(\pi_t^{\text{IDS}})$, it is immediate that $\Delta_t^* \leqslant \sqrt{\lambda g_t^*}$. Therefore

$$\begin{aligned}\mathbb{E}[\Delta_T^*] &\overset{(a)}{\leqslant} \frac{1}{T}\mathbb{E}\sum_{t=1}^{T}\Delta_t^* \leqslant \frac{\sqrt{\lambda}}{T}\mathbb{E}\sum_{t=1}^{T}\sqrt{g_t^*} \\ &\overset{(b)}{\leqslant} \frac{\sqrt{\lambda}}{T}\sqrt{T\mathbb{E}\sum_{t=1}^{T} g_t^*} \overset{(c)}{\leqslant} \sqrt{\frac{\lambda H(\alpha_1)}{T}}.\end{aligned}$$

Inequality (a) uses Equation (10) in the proof of Proposition 8, (b) uses the Cauchy-Schwartz inequality, and (c) follows as in the proof of Proposition 1. □

9.2. Using Information Gain About θ

Information-directed sampling optimizes a single-period objective that balances earning high immediate reward and acquiring information. Information is quantified using the mutual information between the true optimal action A^* and the algorithm's next observation $Y_{t,a}$. In this subsection, we will consider an algorithm that instead quantifies the amount learned through selecting an action a using the mutual information $I_t(\theta; Y_{t,a})$ between the algorithm's next observation and the unknown parameter θ. As highlighted in Section 4.3.2, such an algorithm could invest in acquiring information that is irrelevant to the decision problem. However, in some cases, such an algorithm can be computationally simple while offering reasonable statistically efficiency.

We introduce a modified form of the information ratio

$$\Psi_t^\theta(\pi) := \frac{\Delta_t(\pi)^2}{\sum_{a\in\mathcal{A}} \pi(a) I_t(\theta; Y_{t,a})}, \tag{11}$$

which replaces the expected information gain about A^*, $g_t(\pi) = \sum_{a\in\mathcal{A}} \pi(a) I_t(A^*; Y_{t,a})$, with the expected information gain about θ.

Proposition 10. *For any action sampling distribution* $\tilde{\pi} \in \mathcal{D}(\mathcal{A})$,

$$\Psi_t^\theta(\tilde{\pi}) \leqslant \Psi_t(\tilde{\pi}). \tag{12}$$

Furthermore, if Θ *is finite, and there is some* $\lambda \in \mathbb{R}$ *and policy* $\pi = (\pi_1, \pi_2, \dots)$ *satisfying* $\Psi_t^\theta(\pi_t) \leqslant \lambda$ *almost surely, then*

$$\mathbb{E}[\text{Regret}(T,\pi)] \leqslant \sqrt{\lambda H(\theta) T}. \tag{13}$$

Equation (12) relies on the inequality $I_t(A^*; Y_{t,a}) \leqslant I_t(\theta; Y_{t,a})$, which itself follows from the data processing inequality of mutual information because A^* is a function of θ. The proof of the second part of the proposition is almost identical to the proof of Proposition 1, and is omitted.

We have provided several bounds on the information ratio of π^{IDS} of the form $\Psi_t(\pi_t^{\text{IDS}}) \leqslant \lambda$. By this proposition, such bounds imply that if $\pi = (\pi_1, \pi_2, \dots)$ satisfies

$$\pi_t \in \arg\min_{\pi\in\mathcal{D}(\mathcal{A})} \Psi_t^\theta(\pi)$$

then, $\Psi_t^\theta(\pi_t) \leqslant \Psi_t^\theta(\pi_t^{\text{IDS}}) \leqslant \Psi_t(\pi_t^{\text{IDS}}) \leqslant \lambda$, and the regret bound (13) applies.

9.3. A Tunable Version of Information-Directed Sampling

In this section, we present an alternative form of information-directed sampling that depends on a tuning parameter $\lambda \in \mathbb{R}$. As λ varies, the algorithm strikes a different balance between exploration and exploration. The following proposition provides regret bounds for this algorithm provided λ is sufficiently large.

Proposition 11. *Fix any* $\lambda \in \mathbb{R}$ *such that* $\Psi_t(\pi_t^{\text{IDS}}) \leqslant \lambda$ *almost surely for each* $t \in \{1,\dots,T\}$. *If* $\pi = (\pi_1, \pi_2, \dots)$ *is defined so that*

$$\pi_t \in \arg\min_{\pi\in\mathcal{D}(\mathcal{A})} \left\{\rho(\pi) := \Delta_t(\pi)^2 - \lambda g_t(\pi)\right\}, \tag{14}$$

then

$$\mathbb{E}[\text{Regret}(T,\pi)] \leqslant \sqrt{\lambda H(\alpha) T}.$$

Proof. We have that

$$\rho(\pi_t) \overset{(a)}{\leqslant} \rho(\pi_t^{\text{IDS}}) \overset{(b)}{\leqslant} 0,$$

where (a) follows since π_t^{IDS} is feasible for the optimization problem (14), and (b) follows since

$$0 = \Delta_t(\pi_t^{\text{IDS}})^2 - \Psi_t(\pi_t^{\text{IDS}}) g_t(\pi_t^{\text{IDS}}) \geqslant \Delta_t(\pi_t^{\text{IDS}})^2 - \lambda g_t(\pi_t^{\text{IDS}}).$$

Since $\rho_t(\pi_t) \leqslant 0$, it must be the case that $\lambda \geqslant \Delta_t(\pi_t)^2 / g_t(\pi_t) \overset{\text{Def}}{=} \Psi_t(\pi_t)$. The result then follows by applying Proposition 1. □

Acknowledgments

The authors thank the anonymous referees for feedback and stimulating exchanges, Junyang Qian for correcting miscalculations in an earlier draft pertaining to Examples 3 and 4, and Eli Gutin and Yashodan Kanoria for helpful suggestions.

Endnotes

[1] In their formulation, the reward from selecting action a is $\sum_{i\in a} X_{t,i}$, which is m times larger than in our formulation. The lower bound

stated in their paper is therefore of order $\sqrt{mdT}$. They do not provide a complete proof of their result, but note that it follows from standard lower bounds in the bandit literature. In the proof of Theorem 5 in that paper, they construct an example in which the decision maker plays m bandit games in parallel, each with d/m actions. Using that example, and the standard bandit lower bound (see Theorem 3.5 of Bubeck and Cesa-Bianchi 2012), the agent's regret from each component must be at least $\sqrt{(d/m)T}$, and hence her overall expected regret is lower bounded by a term of order $m\sqrt{(d/m)T} = \sqrt{mdT}$.

[2] Some details related to the derivation of this fact when $Y_{t,a}$ is a general random variable can be found in the appendix of Russo and Van Roy (2016).

[3] Their theoretical guarantees require choosing a somewhat higher quantile, but the authors suggest choosing this quantile, and use it in their own numerical experiments.

[4] We set $\delta = 0$ in the definition of β_t, as this choice leads to a lower value of β_t and stronger performance.

[5] Regret analysis of GP-UCB can be found in (Srinivas et al. 2012). Regret bounds for Thompson sampling can be found in (Agrawal and Goyal 2013b; Russo and Van Roy 2014b, 2016).

References

Abbasi-Yadkori Y, Pál D, Szepesvári C (2011) Improved algorithms for linear stochastic bandits. Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira FCN, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 24 (Curran Associates, Red Hook, NY), 2313–2320.

Abbasi-Yadkori Y, Pál D, Szepesvári C (2012) Online-to-confidence-set conversions and application to sparse stochastic bandits. *Proc. 15th Internat. Conf. Artificial Intelligence Statist. AISTATS '12* (JMLR.org), 1–9.

Agrawal R, Teneketzis D, Anantharam V (1989a) Asymptotically efficient adaptive allocation schemes for controlled iid processes: Finite parameter space. *IEEE Trans. Automatic Control* 34(3): 258–267.

Agrawal R, Teneketzis D, Anantharam V (1989b) Asymptotically efficient adaptive allocation schemes for controlled Markov chains: Finite parameter space. *IEEE Trans. Automatic Control* 34(12): 1249–1259.

Agrawal S, Goyal N (2013a) Further optimal regret bounds for Thompson sampling. *Proc. Sixteenth Internat. Conf. Artificial Intelligence Statist. AISTATS '13*, (JMLR.org), 99–107.

Agrawal S, Goyal N (2013b) Thompson sampling for contextual bandits with linear payoffs. *Proc. 30th Internat. Conf. Machine Learning, ICML '13*, (JMLR.org), 127–135.

Audibert JY, Bubeck S (2009) Minimax policies for adversarial and stochastic bandits. *Proc. 22nd Annual Conf. Learning Theory, COLT '09*, 217–226.

Audibert JY, Bubeck S, Lugosi G (2014) Regret in online combinatorial optimization. *Math. Oper. Res.* 39(1):31–45

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learn.* 47(2):235–256.

Bai A, Wu F, Chen X (2013) Bayesian mixture modelling and inference based Thompson sampling in Monte-Carlo tree search. Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 26 (Curran Associates, Red Hook, NY), 1646–1654.

Bartók G, Foster DP, Pál D, Rakhlin A, Szepesvári C (2014) Partial monitoring-classification, regret bounds, and algorithms. *Math. Oper. Res.* 39(4):967–997.

Boyd S, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press, Cambridge, UK).

Brochu E, Cora V, de Freitas N (2009) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-23, Department of Computer Science, University of British Columbia.

Broder J, Rusmevichientong P (2012) Dynamic pricing under a general parametric choice model. *Oper. Res.* 60(4):965–980.

Bubeck S, Cesa-Bianchi N (2012) Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learn.* 5(1):1–122.

Bubeck S, Eldan R (2016) Multi-scale exploration of convex functions and bandit convex optimization. Feldman V, Rakhlin A, Shamir O, eds. *Proc. 29th Annual Conf. Learn. Theory, COLT '16*, (JMLR.org), 583–589.

Bubeck S, Dekel O, Koren T, Peres Y (2015) Bandit convex optimization: $\sqrt{T}$ regret in one dimension. *Proc. 28th Annual Conf. Learn. Theory, COLT '15* (JMLR.org), 266–278.

Bubeck S, Munos R, Stoltz G, Szepesvári C (2011) X-armed bandits. *J. Machine Learn. Res.* 12:1655–1695.

Cappé O, Garivier A, Maillard OA, Munos R, Stoltz G (2013) Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Ann. Statist.* 41(3):1516–1541.

Chaloner K, Verdinelli I (1995) Bayesian experimental design: A review. *Statist. Sci.* 10(3):273–304.

Chapelle O, Li L (2011) An empirical evaluation of Thompson sampling. Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira FCN, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 24 (Curran Associates, Red Hook, NY), 2249–2257.

Contal E, Perchet V, Vayatis: N (2014) Gaussian process optimization with mutual information. *Proc. 31st International Conf. Machine Learning, ICML '14*, (JMLR.org), 253–261.

Dani V, Hayes T, Kakade S (2008) Stochastic linear optimization under bandit feedback. Servedio RA, Zhang T, eds. *Proc. 21st Annual Conf. Learning Theory, COLT '08*, (Omnipress, Madison, WI), 355–366.

Dani V, Kakade S, Hayes T (2007) Platt JC, Koller D, Singer Y, Roweis ST, eds. The price of bandit information for online optimization. *Advances in Neural Information Processing Systems*, 20 (Currant Associates, Red Hook, NY), 345–352.

Filippi S, Cappé O, Garivier A, Szepesvári C (2010) Parametric bandits: The generalized linear case. Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, eds. *Advances in Neural Information Processing Systems*, Vol. 23 (Currant Associates, Red Hook, NY), 586–594.

Francetich A, Kreps DM (2016a) Choosing a good toolkit, I: Formulation, heuristics, and asymptotic properties. Preprint.

Francetich A, Kreps DM (2016b) Choosing a good toolkit, II: Simulations and conclusions. Preprint.

Frazier P, Powell W (2010) Paradoxes in learning and the marginal value of information. *Decision Anal.* 7(4):378–403.

Frazier P, Powell W, Dayanik S (2008) A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.* 47(5):2410–2439.

Gittins J, Glazebrook K, Weber R (2011) *Multi-Armed Bandit Allocation Indices* (John Wiley & Sons, Hoboken, NJ).

Golovin D, Krause A (2011) Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artificial Intelligence Res.* 42(1):427–486.

Golovin D, Krause A, Ray D (2010) Near-optimal Bayesian active learning with noisy observations. Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, eds. *Advances in Neural Information Processing Systems*, Vol. 23 (Currant Associates, Red Hook, NY), 766–774.

Gopalan A, Mannor S, Mansour Y (2014) Thompson sampling for complex online problems. *Proc. 31st Internat. Conf. Machine Learning* (JLMR.org), 100–108.

Graves T, Lai T (1997) Asymptotically efficient adaptive choice of control laws in controlled Markov chains. *SIAM J. Control Optim.* 35(3):715–743.

Gray R (2011) *Entropy and Information Theory* (Springer, New York).

Hennig P, Schuler C (2012) Entropy search for information-efficient global optimization. *J. Machine Learn. Res.* 13(1):1809–1837.

Hernández-Lobato JM, Hoffman MW, Ghahramani Z (2014) Predictive entropy search for efficient global optimization of black-box functions. Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, 27 (MIT Press, Cambridge, MA), 918–926.

Hernández-Lobato D, Hernández-Lobato JM, Shah A, Adams RP (2016) Predictive entropy search for multi-objective Bayesian optimization. Balcan M-F, Weinberger KQ, eds. *Proc. 33rd Internat. Conf. Machine Learning, ICML '16* (JMLR.org), 1492–1501.

Hernández-Lobato JM, Gelbart MA, Hoffman MW, Adams RP, Ghahramani Z (2015) Predictive entropy search for Bayesian optimization with unknown constraints. Bach FR, Blei DM, eds. *Proc. 32nd Internat. Conf. Machine Learning, ICML '15* (JMLR.org), 1699–1707.

Jaksch T, Ortner R, Auer P (2010) Near-optimal regret bounds for reinforcement learning. *J. Machine Learn. Res.* 11:1563–1600.

Jedynak B, Frazier P, Sznitman R, et al. (2012) Twenty questions with noise: Bayes optimal policies for entropy loss. *J. Appl. Probab.* 49(1):114–136.

Kamiński B (2015) Refined knowledge-gradient policy for learning probabilities. *Oper. Res. Lett.* 43(2):143–147.

Kaufmann E, Cappé O, Garivier A (2012a) On Bayesian upper confidence bounds for bandit problems. *Proc. 15th Internat. Conf. Artificial Intelligence Statist., AISTATS '12* (JMLR.org), 592–600.

Kaufmann E, Korda N, Munos R (2012b) Thompson sampling: An asymptotically optimal finite time analysis. *Proc. 23th Internat. Conf. Algorithmic Learning Theory, ALT '12* (Springer, Berlin), 199–213.

Kleinberg R, Slivkins A, Upfal E (2008) Multi-armed bandits in metric spaces. *Proc. 40th ACM Sympos. Theory Comput., STOC '08* (ACM, New York), 681–690.

Kocsis L, Szepesvári C (2006) Bandit based Monte-Carlo planning. *Proc. 17th Eur. Conf. Machine Learning, ECML '06* (Springer, Berlin), 282–293.

Kushner H (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Engrg.* 86(1):97–106.

Lai T (1987) Adaptive treatment allocation and the multi-armed bandit problem. *Ann. Statist.* 15(3):1091–1114.

Lai T, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* 6(1):4–22.

Lindley DV (1956) On a measure of the information provided by an experiment. *Ann. Math. Statist.* 78(4):986–1005.

Mockus J, Tiesis V, Zilinskas A (1978) The application of Bayesian methods for seeking the extremum. *Towards Global Optim.* 2(2): 117–129.

Niño-Mora J (2011) Computing a classic index for finite-horizon bandits. *INFORMS J. Comput.* 23(2):254–267.

Osband I, Russo D, Van Roy B (2013) (More) efficient reinforcement learning via posterior sampling. Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 26 (Curran Associates, Red Hook, NY).

Piccolboni A, Schindelhauer C (2001) Discrete prediction games with arbitrary feedback and loss. Helmbold D, Williamson B, eds. *Proc. 14th Internat. Conf. Comput. Learning Theory, COLT '01* (Springer, Berlin), 208–223.

Powell W, Ryzhov I (2012) *Optimal Learning*, Vol. 841 (John Wiley & Sons, Hoboken, NJ).

Rusmevichientong P, Tsitsiklis J (2010) Linearly parameterized bandits. *Math. Oper. Res.* 35(2):395–411.

Rusmevichientong P, Shen ZJM, Shmoys D (2010) Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Oper. Res.* 58(6):1666–1680.

Russo D, Van Roy B (2013) Eluder dimension and the sample complexity of optimistic exploration. Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 26 (Curran Associates, Red Hook, NY), 2256–2264.

Russo D, Van Roy B (2014a) Learning to optimize via information-directed sampling. Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, Vol. 27 (Curran Associates, Red Hook, NY), 1583–1591.

Russo D, Van Roy B (2014b) Learning to optimize via posterior sampling. *Math. Oper. Res.* 39(4):1221–1243.

Russo D, Van Roy B (2016) An information-theoretic analysis of Thompson sampling. *J. Machine Learn. Res.* 17(68):1–30.

Russo D, Tse D, Van Roy B (2017) Time-sensitive bandit learning and satisficing Thompson sampling. Preprint arXiv:1704.09028.

Ryzhov I, Frazier P, Powell W (2010) On the robustness of a one-period look-ahead policy in multi-armed bandit problems. *Procedia Comput. Sci.* 1(1):1635–1644.

Ryzhov I, Powell W, Frazier P (2012) The knowledge gradient algorithm for a general class of online learning problems. *Oper. Res.* 60(1):180–195.

Sauré D, Zeevi A (2013) Optimal dynamic assortment planning with demand learning. *Manufacturing Service Oper. Management* 15(3):387–404.

Scott S (2010) A modern Bayesian look at the multi-armed bandit. *Appl. Stochastic Models Bus. Indust.* 26(6):639–658.

Srinivas N, Krause A, Kakade S, Seeger M (2012) Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Trans. Inform. Theory* 58(5):3250–3265.

Valko M, Carpentier A, Munos R (2013a) Stochastic simultaneous optimistic optimization. *Proc. 30th Internat. Conf. Machine Learning, ICML '13* (JMLR.org), 19–27.

Valko M, Korda N, Munos R, Flaounas I, Cristianini N (2013b) Finite-time analysis of kernelised contextual bandits. Nicholson A, Smyth P, eds. *Proc. 29th Conf. Uncertainty in Artificial Intelligence, UAI '13* (AUAI Press, Corvallis, OR), 654–663.

Villemonteix J, Vazquez E, Walter E (2009) An informational approach to the global optimization of expensive-to-evaluate functions. *J. Global Optim.* 44(4):509–534.

Waeber R, Frazier P, Henderson S (2013) Bisection search with noisy responses. *SIAM J. Control Optim.* 51(3):2261–2279.

Daniel Russo is an assistant professor of managerial economics, decision science, and operations at Northwestern University's Kellogg School of Management. His research lies at the intersection of statistical machine learning and sequential decision making, and he contributes to the fields of online optimization and reinforcement learning.

Benjamin Van Roy is a professor of electrical engineering, management science and engineering, and, by courtesy, computer science, at Stanford University, where he has served on the faculty since 1998. His research focuses on understanding how an agent interacting with a poorly understood environment can learn over time to make effective decisions.