The value of information in multi-armed bandits with exponentially distributed rewards

Ilya O. Ryzhov, Warren B. Powell

Operations Research and Financial Engineering,
Princeton University,
Princeton, NJ 08544, USA

Abstract

We consider a class of multi-armed bandit problems where the reward obtained by pulling an arm is drawn from an exponential distribution whose parameter is unknown. A Bayesian model with independent gamma priors is used to represent our beliefs and uncertainty about the exponential parameters. We derive a precise expression for the marginal value of information in this problem, which allows us to create a new knowledge gradient (KG) policy for making decisions. The policy is practical and easy to implement, making a case for value of information as a general approach to optimal learning problems with many different types of learning models.

Keywords: multi-armed bandit, knowledge gradient, optimal learning, exponential rewards

1. Introduction

The multi-armed bandit problem is a fundamental model in optimal learning, the study of efficient information collection. It is an example of a multi-stage stochastic optimization problem subject to environmental uncertainty, that is, the decision-maker’s own uncertainty about key problem parameters. We have $M$ independent reward processes (known as “arms” or “alternatives”) with stationary, yet unknown means. At a particular point in time, we can collect a reward from a single process of our choosing. The payoff can be used to update a statistical estimate of the mean reward of the process, thus providing us with information that can be used to make better decisions in the future. Our goal is to maximize the expected total reward collected over a given (possibly infinite) time horizon. The environmental uncertainty inherent in the problem (our decisions depend on our statistical estimates, which may be inaccurate) leads to the challenge of exploration vs. exploitation. We may wish to choose a process that does not have the highest mean according to our model, simply to see whether our model is accurate.

In order to obtain practical decision-making strategies, it is often necessary to make additional assumptions on the structure of the random rewards. For example, many studies [1, 2] make the assumption that the rewards follow Bernoulli distributions with unknown success probabilities. This model can be motivated by the problem of clinical trials, where an experimental drug can be either successful or unsuccessful in curing a disease. In problems where the rewards are continuous, the Gaussian distribution can provide modeling flexibility and computational advantages.
However, there are numerous areas of application where the payoffs are inherently positive, and thus make an uneasy fit for the Gaussian framework. Some examples include:

1. **Network routing.** We are submitting queries to DNS servers. Any server can resolve any query, but each server has a different, unknown expected latency. Our goal is to resolve all queries as quickly as possible, within a finite, small time horizon (the total number of queries). The time cost of a query is always positive.

2. **Multi-skill call centers.** We are assigning calls arriving in series to technicians. The payoff is the time needed by a technician to resolve the issue. The objective is to minimize the total time needed to finish all jobs. The total number of jobs can potentially be large.

3. **Supply chain management.** We have a choice of suppliers when contracting for a service or product. Our expected inventory costs under each supplier depend on factors such as the supplier’s service reliability, which can only be learned by doing business with the supplier for a period of time. Our goal is to find the best supplier while minimizing total inventory costs incurred over time.

The classical approach to bandit problems uses index policies to make decisions. For each arm, we first compute an index, which is a number that depends only on our beliefs about that particular arm, independently of what we might believe about the other arms. We then choose the arm with the largest index. Thus, we decompose the problem into $M$ individual computations. The work by [3] showed that, if the reward processes are independent, then the optimal decision-making strategy is an index policy, although computing the “Gittins indices” may be difficult.

Index policies are especially powerful in the case of Gaussian rewards. In a Gaussian model, the Gittins index for an arm can be written as a linear function of a “standard Gittins” index that does not depend on our estimate of the mean reward [4], similar to how Gaussian noise can be viewed as a scaling of a standard Gaussian random variable. Furthermore, Gittins indices are easier to approximate in the Gaussian case, via an analogy to continuous Gaussian processes [5, 6, 7]. Other index policies for Gaussian rewards can be found in [8] and [9].

At the same time, there are fewer computable, practical index policies for the exponential model. Scaling properties are derived in [10], and [4] provides tables of Gittins indices for particular values of our statistical estimates, but does not give numbers for every possible situation. The seminal work by [5] on approximating Gittins indices suggests using the Gaussian approximation as a heuristic for non-Gaussian problems. One index policy derived specifically for exponential rewards, based on the upper confidence bound approach of [11], can be found in [12]. However, even this policy relies on parameters that are difficult to compute optimally [13]. Outside of this work, [14, 15] provide simple heuristics, such as the $\epsilon$-greedy and soft-max methods, that can be applied to general multi-stage learning problems.

Non-Gaussian distributions have appeared in the broader optimal learning literature. For example, [16] considers a newsvendor problem where the demand follows an exponential distribution with an unknown parameter, whereas [17] studies a dynamic pricing problem with an unknown Poisson arrival rate. While the optimization aspect of these problems is more complex than for multi-armed bandits (e.g. maximizing a newsvendor payoff), they contain a single unknown parameter. This makes the problems somewhat more tractable, with [16] even finding a computable optimal policy. Optimization in the bandit problem is relatively straightforward, but there are many unknown parameters.

In this paper, we use the knowledge gradient (KG) approach to derive a computable policy for bandit problems with exponential rewards. KG is based on the idea of value of information, first applied in optimal learning by [18, 19] to ranking and selection, an offline problem where we only estimate the highest mean reward, instead of maximizing total reward collected over time. A more theoretical treatment can be found in [20], with extensions pursued in [21, 22]. The KG method quantifies the effect of a single measurement on our estimate of the best reward. Essentially, where index policies consider each arm separately, but over a long time horizon, the KG method considers the evolution of our beliefs about all the arms, but only one time step into the future. This one step often carries a great deal of information, although there are problems where the value of information is non-concave [23].

The KG approach has been applied to multi-armed bandits before [24, 25, 26], but thus far, only Gaussian and Bernoulli models have been considered. However, in both of those settings, KG yielded computable, practical decision-making strategies that performed robustly [27] in experiments. We show that both advantages are retained in the exponential model: we obtain a closed-form expression for the value of information in this setting, and we show that it outperforms other index policies and heuristics. We emphasize the way in which the general KG concept, when applied specifically to the exponential problem, yields a computable strategy tailored to the problem structure, suggesting that KG can serve as a general methodology for creating computable policies in many classes of optimal learning problems.
2. Multi-armed bandits with exponential rewards

Suppose that the random payoff of arm $x$, for $x = 1, ..., M$, follows an exponential distribution with parameter $\lambda_x$. This parameter is unknown, but we use a Bayesian prior distribution to represent our beliefs about it. We assume that $\lambda_x \sim \text{Gamma}(\alpha_x^0, \beta_x^0)$, with $\alpha_x^0 > 1$ and $\beta_x^0 > 0$. Thus, our initial estimate of $\lambda_x$ is $E\lambda_x = \frac{\beta_x^0}{\alpha_x^0 - 1}$, our initial estimate of the mean reward is $E\frac{1}{\lambda_x} = \frac{\beta_x^0}{\alpha_x^0}$, and our uncertainty about the accuracy of this estimate is encoded by the gamma distribution. In particular, if $\beta_x^0$ is very large relative to $\alpha_x^0$, the variance of the gamma distribution will be small, implying that we have a high degree of confidence in our estimate.

We say that an event happens “at time $n$” if it occurs after we have made $n$ decisions, but before we have made the $(n + 1)$st. Denote by $x^n \in \{1, ..., M\}$ the $(n + 1)$st arm pulled. Then, we collect a reward $\hat{Y}_{x}^{n+1} \sim \text{Exp}(\lambda_x)$, drawn i.i.d. from the true underlying distribution. Our posterior distribution of belief at time $n$ is $\text{Gamma}(\alpha_x^n, \beta_x^n)$. The parameters of the gamma distribution evolve according to simple recursive equations [28]. The shape parameter changes deterministically as follows:

$$\alpha_x^{n+1} = \begin{cases} \alpha_x^n + 1 & \text{if } x^n = x \\ \alpha_x^n & \text{otherwise.} \end{cases}$$

The updating equation for the scale parameter is

$$\beta_x^{n+1} = \begin{cases} \beta_x^n + \hat{Y}_{x}^{n+1} & \text{if } x^n = x \\ \beta_x^n & \text{otherwise.} \end{cases}$$

Let $\alpha^n = (\alpha_1^n, ..., \alpha_M^n)$ and $\beta^n = (\beta_1^n, ..., \beta_M^n)$. The knowledge state $s^n = (\alpha^n, \beta^n)$ completely characterizes our beliefs about all the alternatives at time $n$, and (1-2) describe the way in which our beliefs evolve over time as we make decisions.

Our goal is to create a policy that will make decisions adaptively. Let $N$ denote the time horizon, or the total number of decisions we are allowed to make. We allow $N$ to be infinite. We define a policy $\pi$ to be a sequence $(X^{\pi,n})_{n=0}^{\infty}$ of decision rules, where each decision rule $X^{\pi,n}$ is a function mapping a knowledge state $s^n$ to an alternative $X^{\pi,n}(s^n) \in \{1, ..., M\}$. Thus, by specifying a policy, we automatically lock in a way to make decisions in every possible situation. Our objective is to find the policy that achieves the biggest expected reward, or

$$\max_{\pi} E\sum_{n=0}^{N} \frac{\gamma^n Y_{X^{\pi,n}(s^n)}}{\lambda_{X^{\pi,n}(s^n)}},$$

for some discount factor $0 < \gamma \leq 1$. Here, $E\gamma^n$ is an expectation over the joint distribution of the true parameters $\lambda_x$, and the random rewards $Y_{x}^{n+1}$ received at each time step, given that all decisions are made according to the policy $\pi$.

3. The value of information and the knowledge gradient policy

Although we place our Bayesian priors on the unknown exponential parameters $\lambda_x$, the average reward collected from arm $x$ is the reciprocal $\frac{1}{\lambda_x}$. Our estimate of this quantity at time $n$ is the expectation over the prior distribution, $E\frac{1}{\lambda_x} = \frac{\beta_x^n}{\alpha_x^n - 1}$. Note that this is different from the point estimate of $\frac{\beta_x^n}{\alpha_x^n}$ that we would obtain if we were to assume that $\lambda_x$ is equal to our estimate of $\frac{\beta_x^n}{\alpha_x^n}$. When we take our uncertainty about $\lambda_x$ into consideration, by taking an expectation over the prior distribution, we reach a different estimate.

Suppose that we choose alternative $x$ at time $n$. The value of information obtained from this decision is defined to be the expected improvement in our beliefs about the largest possible reward,

$$I_{X}^{n+1} = E_x\left[\left(\max_y \frac{\beta_y^{n+1}}{\alpha_y^{n+1} - 1} - \left(\max_y \frac{\beta_y^{n}}{\alpha_y^n - 1}\right)\right)\right].$$

If our beliefs were fixed starting at time $n$, with no chance to learn any new information, we would choose the arm with the highest estimated expected reward $\max_y \frac{\beta_y^{n}}{\alpha_y^n - 1}$. The right-hand side of (3) represents the amount by which this
estimate improves between time \( n \) and time \( n + 1 \), as a result of our decision to choose \( x \) at time \( n \). The expectation \( E^x \) is conditional given \( s^n \) and \( x^n = x \). Because (3) is written as a difference, we refer to \( y_{x,n}^{KG,n} \) as the knowledge gradient or the KG factor of alternative \( x \) at time \( n \). In this section, we give a closed-form expression for the knowledge gradient, and show how it can be used to create a simple and computationally efficient policy for making decisions.

### 3.1. Computation of the knowledge gradient

Given \( s^n \) and \( x^n = x \), we can compute the conditional distribution of \( \hat{y}_{x}^{n+1} \) as

\[
P^n(\hat{y}_{x}^{n+1} > y) = E^n P^n(\hat{y}_{x}^{n+1} > y | Y_n),
\]

\[
= E^n e^{-\lambda_y},
\]

\[
= \left( \frac{\beta_x^n}{\beta_x^n + y} \right)^{\alpha_x^n} \quad \text{for } y > 0.
\]

Consequently,

\[
P^n(\beta_x^n + \hat{y}_{x}^{n+1} > y) = \left( \frac{\beta_x^n}{y} \right)^{\alpha_x^n} \quad \text{for } y > \beta_x^n,
\]

which means that the conditional distribution of \( \beta_x^n + \hat{y}_{x}^{n+1} \) given \( s^n \) and \( x^n = x \), is the Pareto distribution [29] with shape parameter \( \alpha_x^n \) and scale parameter \( \beta_x^n \). As a result, we can write

\[
E^n \max_y \frac{\beta_x^n + \hat{y}_{x}^{n+1}}{\alpha_x^n} = E \max \left\{ \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1}, \frac{Y}{\alpha_x^n} \right\}
\]

(4)

with \( Y \sim \text{Pareto}(\alpha_x^n, \beta_x^n) \). Jensen’s inequality implies that

\[
E \max \left\{ \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1}, \frac{Y}{\alpha_x^n} \right\} \geq \max \left\{ \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1}, \frac{E Y}{\alpha_x^n} \right\} = \max \left\{ \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1}, \frac{\beta_x^n}{\alpha_x^n - 1} \right\} = \max \frac{\beta_x^n}{\alpha_x^n - 1}.
\]

Whence \( y_{x,n}^{KG,n} \geq 0 \). In other words, the value of information is always positive, regardless of which alternative we choose.

We now present the main result of this section: a closed-form expression for the KG factor.

**Theorem 3.1.** Let \( C_x^n = \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1} \). The KG factor of alternative \( x \) at time \( n \) is given by

\[
y_{x,n}^{KG,n} = \begin{cases} 
\frac{1}{(\alpha_x^n - 1)(C_x^n)^{\alpha_x^n - 1}} \left( \frac{\beta_x^n}{\alpha_x^n} \right)^{\alpha_x^n} - \left( \frac{\beta_x^n}{\alpha_x^n - 1} \right) & \text{if } \frac{\beta_x^n}{\alpha_x^n - 1} \leq C_x^n \\
0 & \text{if } \frac{\beta_x^n}{\alpha_x^n - 1} > C_x^n
\end{cases}
\]

(5)

**Proof:** We compute the right-hand side of (3) in several cases. First, consider the case where

\[
\frac{\beta_x^n}{\alpha_x^n} > C_x^n.
\]

Because \( Y \geq \beta_x^n \) by the definition of a Pareto distribution [28], it follows that

\[
E \max \left\{ C_x^n, \frac{Y}{\alpha_x^n} \right\} = E \frac{Y}{\alpha_x^n} = \frac{\beta_x^n}{\alpha_x^n - 1}.
\]

Since \( \frac{\beta_x^n}{\alpha_x^n - 1} > \frac{\beta_x^n}{\alpha_x^n} \geq C_x^n \), we know that \( \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1} = \frac{\beta_x^n}{\alpha_x^n - 1} \), and therefore,

\[
E^n \left( \max_{y \neq x} \frac{\beta_x^n + \hat{y}_{x}^{n+1}}{\alpha_x^n + 1} - \left( \max_{y \neq x} \frac{\beta_x^n}{\alpha_x^n - 1} \right) \right) = 0.
\]
In the case where $\frac{\beta^n}{\alpha^n} \leq C^n_x$, we find
\[
\mathbb{E} \max \left\{ \frac{C^n_x Y}{\alpha^n_x} \right\} = C^n_x \cdot P(Y \leq \alpha^n_x C^n_x) + \int_{\alpha^n_x C^n_x}^{\infty} \frac{y}{\alpha^n_x} \frac{\alpha^n_x}{\alpha^n_x + \beta^n} dy
\]
\[
= C^n_x \left( 1 - \left( \frac{\beta^n}{\alpha^n_x C^n_x} \right)^{\alpha^n_x} \right) + \left( \frac{\beta^n}{\alpha^n_x (\alpha^n_x + 1)} \right)^{\alpha^n_x} y^{\alpha^n_x - 1} C^n_x \alpha^n_x
\]
\[
= C^n_x + \frac{(\beta^n)^{\alpha^n_x}}{\alpha^n_x C^n_x} \left( 1 + \frac{1}{\alpha^n_x - 1} \right) \frac{\beta^n}{\alpha^n_x}
\]
\[
= C^n_x + \frac{1}{\alpha^n_x - 1} \left( \frac{\beta^n}{\alpha^n_x} \right)^{\alpha^n_x - 1}
\]
Depending on whether or not $\frac{\beta^n}{\alpha^n_x - 1} \leq C^n_x$, we obtain the corresponding case of (5).

We see that the alternative argument is penalized by the KG calculation. It is possible for our estimate of a particular reward to be so far ahead of other rewards that choosing the same alternative again will not change our estimate of the best reward, regardless of the outcome of the random observation. Furthermore, even if an alternative is ahead of the others, but still sufficiently close to the second-best, we subtract a positive penalty term from its KG factor. On the other hand, the information obtained by choosing an alternative that seems to be suboptimal is always strictly positive; there is always a chance that we will be surprised by the outcome of our decision.

Figure 1 presents a numerical illustration of the KG formula as a function of $C^n_x$, for fixed values of $\alpha^n_x$ and $\beta^n_x$. The maximum is achieved at $C^n_x = \frac{\beta^n_x}{\alpha^n_x - 1}$, that is, when the top two alternatives are tied. If this is not the case, the KG quantity declines more slowly as $C^n_x > \frac{\beta^n_x}{\alpha^n_x - 1}$, suggesting that an alternative is favoured if it seems to be suboptimal.

If we are seeking to minimize costs incurred (as in our motivating applications), and not to maximize rewards collected, we define the value of a single decision as
\[
\nu^{\text{kg},n}_x = \mathbb{E} \left( \frac{\min_y \{ \beta^n_y \} \alpha^n_y}{\alpha^n_y - 1} - \frac{\min_y \{ \beta^{n+1}_y \} \alpha^{n+1}_y}{\alpha^{n+1}_y - 1} \right),
\]
because improvement is now associated with reducing our estimate of the smallest cost. Letting $C^n_x = \min_{y \neq x} \frac{\beta^n_y}{\alpha^n_y - 1}$ and repeating the calculations of Theorem 3.1, we will arrive at the same expression (5). However, since $C^n_x$ now refers to
the smallest estimate other than \(x\), the formula now takes on a new meaning. The alternative \(\arg\max_x \frac{\beta^x}{\alpha^x} \) will now be the only alternative that does not receive a penalty. For other alternatives, if their cost is believed to be sufficiently high, choosing them will provide no useful information. Information about an apparently suboptimal alternative can only be valuable if our beliefs about that alternative are sufficiently close to the current best.

This behaviour is due to the asymmetry of the gamma and exponential distributions. The quantity \(\frac{\lambda}{\alpha}\) can become arbitrarily large, depending on the outcome of \(Y\), but not arbitrarily small. Thus, our estimates of the mean rewards tend to become larger. If we want the largest mean reward, this allows us to do more exploration, because measuring the alternative that seems to be the best is less likely to change our beliefs about the argmax. However, if we want the smallest mean cost, measuring the current best alternative will induce sufficient exploration by itself.

3.2. The knowledge gradient policy

Suppose that \(N < \infty\) and \(\gamma = 1\). The KG policy for multi-armed bandit problems, first derived in [24, 26] for Gaussian reward processes, makes decisions according to the rule

\[
X_{KG}^{n} (s^n) = \arg\max_x \frac{\beta^x}{\alpha^x} + (N - n) \gamma_{x}^{KG,n}.
\]

(7)

For each alternative, we consider the sum of its estimated reward \(\frac{\beta^x}{\alpha^x}\) and a value of information term, scaled by the number of time periods remaining. Essentially, the KG policy assumes that the current decision at time \(n\) will be the last to change our beliefs (that is, \(s^n = s^{n+1}\) for all \(n' \geq n + 1\)). The value of the information obtained from this measurement is given by \(\gamma_{x}^{KG,n}\). However, the information will also improve our ability to collect rewards in every time period after \(n + 1\), so we multiply \(\gamma_{x}^{KG,n}\) by a scaling factor representing the number of time periods remaining. If \(\gamma < 1\), our expression in (7) becomes

\[
X_{KG}^{n} (s^n) = \arg\max_x \frac{\beta^x}{\alpha^x} + \gamma \frac{1 - (N - n) \gamma_{x}^{KG,n}}{1 - \gamma}.
\]

and if \(N \to \infty\), we obtain the infinite-horizon policy

\[
X_{KG}^{\infty} (s^n) = \arg\max_x \frac{\beta^x}{\alpha^x} + \gamma \frac{\gamma_{x}^{KG,n}}{1 - \gamma}.
\]

(8)

Lastly, if we are minimizing costs rather than maximizing rewards, the same argument used in [24, 26] can be repeated to obtain a policy

\[
X_{KG}^{n} (s^n) = \arg\min_x \frac{\beta^x}{\alpha^x} + (N - n) \gamma_{x}^{KG,n}.
\]

To our knowledge, multi-armed bandit problems have rarely been considered in a minimization framework, perhaps because there exists a symmetry between minimization and maximization if the rewards are Gaussian or Bernoulli.

4. Experimental results

Two policies \(\pi_1,\pi_2\) can be compared using the performance measure

\[
C^{\pi_1,\pi_2} = \sum_{n=0}^{N} \frac{1}{\lambda^{X_{\pi_1,n}(s^n)}} - \frac{1}{\lambda^{X_{\pi_2,n}(s^n)}}.
\]

(9)

the difference in average objective value collected. To reduce the variance of our performance measure, we compute it using the true parameters \(\lambda\) rather than the actual observed rewards \(\hat{Y}^n\). However, this requires us to know the exact values of \(\lambda\). For this reason, we use a simulation study to validate our policies. We first generated 100 problems, each with a different set of starting priors \((\alpha^0, \beta^0)\). For each problem, we recorded the rewards collected by multiple policies in \(10^4\) simulations. The value of \(\lambda\) was generated independently from the prior \(\text{Gamma}(\alpha^0, \beta^0)\) at the beginning of each simulation, then fixed for the duration of the time horizon. In order for our simulations to cover a wide range of
values for the true parameters, we generated the prior parameters $\alpha^0$ and $\beta^0$ from uniform distributions on the intervals $[2, 3]$ and $[0, 1]$, respectively. Thus, our Bayesian modeling assumption holds in the experiments: the values $\lambda_i$ do indeed come from the prior distributions.

For each problem (that is, each of the 100 sets of priors), we divided the full set $10^4$ simulations into groups of 500 to obtain approximately normal estimates of the objective value $E^x \sum_{n=1}^{N_x} \frac{1}{\lambda_{nx}(\pi^i)}$, then averaged over these groups to obtain estimates of the mean values. Taking differences between these estimates for different choices of $\pi_1$ and $\pi_2$ in turn yields estimates of (9).

We considered an infinite-horizon setting with $\gamma = 0.9$ and $\gamma = 0.99$, and used the version of the KG policy given in (8), with $\nu^0_{KG, x}$ computed as in (5). We also tested several other policies, which are described below.

Approximate Gittins indices (Gitt). The biggest advances in approximating Gittins indices have been made in the setting of Gaussian rewards. In this case, the most advanced approximation from [7] is given by

$$X^{Gitt, n}(s^n) = \arg \max_x \mu^x + \sigma^W_x \sqrt{-\log \gamma} \cdot \tilde{b} \left( \frac{(\sigma^x)^2}{(\sigma^W_x)^2 \log \gamma} \right),$$

where

$$\tilde{b}(s) = \begin{cases} 
\frac{\sqrt{s}}{\sqrt{2}} & s \leq \frac{1}{7} \\
\frac{1}{2} + \frac{1}{4} \log s & \frac{1}{7} < s \leq 100 \\
\sqrt{s(2 \log s - \log \log s - \log 16\pi)^{3}} & s > 100.
\end{cases}$$

The quantities $\mu^x$ and $(\sigma^x)^2$ represent the mean and variance of our beliefs about the mean rewards. The work by [5] recommends using this approximation for the non-Gaussian case, based on a central limit argument. In our problem, the mean reward of arm $x$ is $\frac{\gamma^x}{1 - \lambda^x}$. Taking the mean and variance of this quantity with respect to the prior distribution of $\lambda^x$ yields $\frac{\gamma^x}{\sigma^2 - 1}$ and $\frac{(\sigma^x)^2}{(\sigma^2 - 1)^2 (\sigma^2 - 2)}$. Finally, $(\sigma^W_x)^2$ represents the variance of the random reward received from arm $x$ in each time step. When the rewards are exponential, this quantity is $\frac{1}{\lambda^x}$. The work by [5] also recommends using a point estimate for $(\sigma^W_x)^2$, which in this case is $\frac{(\sigma^x)^2}{(\sigma^2 - 1)^2 (\sigma^2 - 2)}$.

It is important to note that this approximation is subject to an additional source of error in our experiments. In addition to the error inherent in the approximation itself, we are also using the policy in a non-Gaussian setting. At the same time, this approximation is very computationally efficient, yielding a very fast decision rule. It can also be applied to problems with arbitrary values of $\alpha^0$ and $\beta^0$, whereas the exact Gittins indices computed for the exponential setting in [4] only cover a narrow selection of prior values.

Upper confidence bounds (UCB). Most of the work on upper confidence bound policies [11, 8, 9] focuses on either Gaussian reward processes, or rewards with bounded support (e.g. Bernoulli). However, [12] proposes one UCB-type method for the specific case of exponential rewards,

$$X^{UCB, n}(s^n) = \arg \max_x \frac{\beta^x}{\alpha^x - 1} + B \min \left\{ \sqrt{\frac{2 \log n + 2 \log \log n}{N^n_x}}, 1 \right\},$$

where $N^n_x$ is the number of times arm $x$ has been pulled (including the current time), and $B$ is a problem-dependent constant. It has been pointed out by [13] that $B$ is difficult to compute optimally. We treated $B$ as a tunable parameter, and found that $B = 5$ appeared to produce the best results.

<table>
<thead>
<tr>
<th>KG vs.</th>
<th>Gitt</th>
<th>UCB</th>
<th>Eps</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.6301</td>
<td>0.4447</td>
<td>1.2185</td>
<td>0.5458</td>
</tr>
<tr>
<td>Std. error</td>
<td>0.5359</td>
<td>0.5956</td>
<td>0.5779</td>
<td>0.5598</td>
</tr>
</tbody>
</table>

Table 1: Means and standard errors for experiments with $\gamma = 0.9$. 

Ilya O. Ryzhov et al. / Procedia Computer Science 4 (2011) 1363–1372

1369
**Epsilon-greedy (Eps).** The epsilon-greedy heuristic [14] chooses an arm uniformly at random with probability \( \frac{1}{n} \), and chooses the arm given by \( \arg \max_x \frac{\beta_x}{\alpha_x} \) the rest of the time. It is thus a hybrid of a Bayes-greedy policy (explained below), and a random exploration policy.

**Bayes-greedy (Greedy).** The Bayes-greedy policy is given by \( X^{Bayes,n}(\pi) = \arg \max_x \frac{\beta_x}{\alpha_x} \). We use the term “Bayes-greedy” to distinguish this policy, which maximizes an expectation of the mean reward of an arm over our distribution of belief about that arm, from the “point-estimate” heuristic, which assumes that \( \lambda_x \) is equal to its estimated mean \( \frac{\alpha_x}{\beta_x} \) and thus chooses the arm given by \( \arg \max_x \frac{\beta_x}{\alpha_x} \). We found that the Bayes-greedy approach consistently outperformed the point-estimate method, and so we do not report results for the point-estimate policy.

Table 1 reports the mean values of (9), averaged across 100 problems, with KG as \( \pi_1 \) and the other policies as \( \pi_2 \), with a discount factor of \( \gamma = 0.9 \). The last row of the table also gives the standard errors for these estimates. We find that KG significantly outperforms the Gittins and epsilon-greedy methods, and narrowly outperforms UCB and Bayes-greedy on average. However, although the mean difference in performance is not statistically significant, we can get more insight by looking at the empirical distribution of (9) across 100 problems, shown in Figure 2. The labels show the total number of problems for which KG outperformed the competition, so “KG/UCB: 78/100” indicates that KG outperformed the UCB method on 78/100 test problems. We see that KG outperforms both UCB and Bayes-greedy on a majority of problems, with a significant positive tail, that is, a set of problems where the margin of victory for KG is noticeably greater than the margin of loss when KG loses.

Table 2 reports the means and standard errors of (9) for \( \gamma = 0.99 \). Positive values for the means indicate that KG outperformed the competing policy. In this setting, we find that the UCB and Bayes-greedy policies become less competitive, whereas the approximate Gittins policy improves considerably. In Figure 3, we see that this is the only policy to ever outperform KG on our test problems. When the discount factor is high, the effective time horizon is longer; essentially, there are more time periods where we have a chance to collect rewards that are relatively large. In that setting, a policy that does not look ahead at all (the greedy policy) is less effective, whereas a policy that is designed to look over an infinite horizon (as the Gittins policy does) performs better. However, it is interesting to note that KG performs competitively against all policies in both settings, which can be interpreted as a kind of robustness.

<table>
<thead>
<tr>
<th>KG vs.</th>
<th>Gitt</th>
<th>UCB</th>
<th>Eps</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.7307</td>
<td>47.0150</td>
<td>65.5725</td>
<td>63.6352</td>
</tr>
<tr>
<td>Std. err</td>
<td>2.7146</td>
<td>3.5066</td>
<td>4.4008</td>
<td>5.0035</td>
</tr>
</tbody>
</table>

Table 2: Means and standard errors for experiments with \( \gamma = 0.99 \).
Finally, Table 3 reports the average number (across 100 problems) of distinct alternatives measured by each policy. Although each test problem has 100 alternatives, many of them are never measured by any policy. Instead, each policy focuses on a subset of alternatives that seem to be good. The Gittins policy explores more than the KG policy, which is consistent with the results reported in [24]. The difference is especially large for $\gamma = 0.9$, when the Gittins approximation underperforms. When the discount factor is small, it is more important to make good decisions in the early iterations than to explore the choice set, so a policy that does less exploration can be more effective. When the discount factor is large, the gap between KG and Gittins closes, although KG still performs competitively.

We conjecture that the Gittins approximation underperforms in the case where $\gamma = 0.9$ because the total discounted rewards are less similar to Gaussian rewards. Furthermore, the policy of [7] seems to be subject to greater approximation error in the small-discount setting, as evidenced by the experiments in [26]. The development of effective Gittins index approximations, particularly in the case of non-Gaussian rewards, remains a matter for further study.

5. Conclusions

We have derived a knowledge gradient policy for a class of multi-armed bandit problems where the rewards are assumed to be exponentially distributed with unknown parameters. Our policy can be computed using a closed-form expression for the value of information, and does not require any tunable parameters. In addition to its computational convenience, our experiments have shown that it performs competitively against other index policies in the exponential setting. Some of the competing policies are effective for smaller discount factors, while others can be effective for larger discount factors, but KG remains competitive against all of them in both settings. A theoretical analysis of the long-run behaviour of the KG policy, whether in an offline or online setting, remains a subject for future research.

Recent work on optimal learning has sought to break out of the confines of the classic bandit model, whether by considering correlations between alternatives [21, 26], variable time horizons [30], or more sophisticated underlying optimization models [31]. While we have considered a classic bandit model in this paper, we have shown how

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>KG</th>
<th>Gitt</th>
<th>UCB</th>
<th>Eps</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>14.3181</td>
<td>34.0689</td>
<td>21.2023</td>
<td>13.0491</td>
<td>6.4616</td>
</tr>
<tr>
<td>0.99</td>
<td>32.4997</td>
<td>38.0255</td>
<td>33.4260</td>
<td>21.0575</td>
<td>6.4549</td>
</tr>
</tbody>
</table>

Table 3: Average number of distinct alternatives measured by each policy.
the KG logic yields an efficient computational algorithm for the case of exponential rewards, which has received relatively little attention. It is our hope that this work will contribute to the idea that the value of information concept underlying KG is a flexible and broadly applicable methodology that can yield computable decision-making rules in many problem classes.

Acknowledgments

This work was supported in part by AFOSR contract FA9550-08-1-0195 through the Center for Dynamic Data Analysis.