Multi-armed Bandit Algorithms and Empirical Evaluation

Joannès Vermorel[1] and Mehryar Mohri[2]

[1] École normale supérieure, 45 rue d'Ulm, 75005 Paris, France
joannes.vermorel@ens.fr
[2] Courant Institute of Mathematical Sciences,
719 Broadway, New York, NY 10003, USA
mohri@cs.nyu.edu

Abstract. The multi-armed bandit problem for a gambler is to decide which arm of a K-slot machine to pull to maximize his total reward in a series of trials. Many real-world learning and optimization problems can be modeled in this way. Several strategies or algorithms have been proposed as a solution to this problem in the last two decades, but, to our knowledge, there has been no common evaluation of these algorithms.

This paper provides a preliminary empirical evaluation of several multi-armed bandit algorithms. It also describes and analyzes a new algorithm, Poker (Price Of Knowledge and Estimated Reward) whose performance compares favorably to that of other existing algorithms in several experiments. One remarkable outcome of our experiments is that the most naive approach, the ϵ-greedy strategy, proves to be often hard to beat.

1 Introduction

In many real-world situations, decisions are made in order to maximize some expected numerical reward. But decisions, or the actions they generate, do not just bring in more reward, they can also help discover new knowledge that could be used to improve future decisions. Such situations include clinical trials [11] where different treatments need to be experimented with while minimizing patient losses, or adaptive routing efforts for minimizing delays in a network [4]. The questions that arise in all these cases are related to the problem of balancing reward maximization based on the knowledge already acquired and attempting new actions to further increase knowledge, which is known as the exploitation vs. exploration tradeoff in reinforcement learning.

The multi-armed bandit problem, originally described by Robins [19], is an instance of this general problem. A multi-armed bandit, also called K-armed bandit, is similar to a traditional slot machine (one-armed bandit) but in general has more than one lever. When pulled, each lever provides a reward drawn from a distribution associated to that specific lever. Initially, the gambler has no knowledge about the levers, but through repeated trials, he can focus on the most rewarding levers.

This paper considers the *opaque* bandit problem where a unique reward is observed at each round, in contrast with the *transparent* one where all rewards are observed [14]. To our knowledge, there is no empirical comparison for the

J. Gama et al. (Eds.): ECML 2005, LNAI 3720, pp. 437–448, 2005.

transparent bandit problem either. More formally, the opaque stochastic K-armed bandit (*bandit* for short) can be seen as a set of real distributions $\mathcal{B} = \{R_1, \dots, R_K\}$, each distribution being associated to the rewards brought in by a specific lever.[1] Let $\mu_1, \dots, \mu_K$ be the mean values associated to these reward distributions. The gambler plays iteratively one lever at each round and observes the associated reward. His objective is to maximize the sum of the collected rewards. The *horizon* H is the number of rounds that remains to be played. The bandit problem is formally equivalent to a one-state Markov Decision Process (MDP), but the general study of MDPs goes beyond the scope of this paper.

A different version of the bandit problem has been studied by [10, 23, 9, 8] where the reward distributions are assumed to be known to the player. This problem is not about balancing exploration and exploitation, it admits an optimal solution based on the so-called Gittins indices. This paper deals with bandit problems found in practice where the assumption about the prior knowledge of the payoffs typically does not hold (see for example section 4).

The *regret* ρ after T rounds is defined as the difference between the reward sum associated to an optimal strategy and the sum of the collected rewards $\rho = T\mu^* - \sum_{t=1}^{T} \widehat{r}_t$ where μ^* is the maximal reward mean, $\mu^* = \max_k\{\mu_k\}$, and $\widehat{r}_t$ the reward at time t. A strategy whose average regret per round tends to zero with probability 1 for any bandit problem when the horizon tends to infinity is a *zero-regret strategy*. Intuitively, zero-regret strategies are guaranteed to converge to an optimal strategy, not necessarily unique, if enough rounds are played.

The problem of determining the best strategy for the gambler is called the multi-armed bandit problem. Many strategies or algorithms have been proposed as a solution to this problem in the last two decades, but, to our knowledge, there has been no common evaluation of these algorithms. This paper provides the first preliminary empirical evaluation of several multi-armed bandit algorithms. It also describes and analyzes a new algorithm, POKER (Price Of Knowledge and Estimated Reward) whose performance compares favorably to that of other existing algorithms in several experiments.

The paper is organized as follows. We first present an overview of several bandit strategies or algorithms (Section 2), then introduce a new algorithm, POKER (Section 3), and describe our experiments with both an artificially generated dataset and a real networking dataset. The results of an empirical evaluation of several bandit algorithms, including POKER are reported in Section 4.

2 Bandit Algorithms Overview

The exploration vs. exploitation tradeoff is often studied under more general models such as MDPs. We have restricted this overview to methods that apply to the stateless case, specific to the bandit problem. There is, however, a significant amount of literature dealing with MDPs, see [17, 6] for a review. Slowly changing worlds have also been considered in [22, 3].

[1] Several algorithms have also been designed for the non-stochastic bandit problem [3] where much weaker assumptions are made about the levers' rewards, but this paper will focus on the stochastic bandit problem which has been studied the most so far.

2.1 The ϵ-Greedy Strategy and Semi-uniform Variants

ϵ-greedy is probably the simplest and the most widely used strategy to solve the bandit problem and was first described by Watkins [24]. The **ϵ-greedy strategy** consists of choosing a random lever with ϵ-frequency, and otherwise choosing the lever with the highest estimated mean, the estimation being based on the rewards observed thus far. ϵ must be in the open interval $(0, 1)$ and its choice is left to the user. Methods that imply a binary distinction between exploitation (the greedy choice) and exploration (uniform probability over a set of levers) are known as *semi-uniform* methods.

The simplest variant of the ϵ-greedy strategy is what we will refer to as the **ϵ-first strategy**. The ϵ-first strategy consists of doing the exploration all at once at the beginning. For a given number $T \in \mathbb{N}$ of rounds, the levers are randomly pulled during the ϵT first rounds (pure exploration phase). During the remaining $(1-\epsilon)T$ rounds, the lever of highest estimated mean is pulled (pure exploitation phase). Here too, ϵ must be in the open interval $(0, 1)$ and its choice is left to the user. The ϵ-first strategy has been analyzed within the PAC framework by [7] and [16]. Even-Dar et al. show in [7] that a total of $\mathcal{O}\left(\frac{K}{\alpha^2}\log\left(\frac{K}{\delta}\right)\right)$ random pulls suffices to find an α-optimal arm with probability at least $1-\delta$. This result could be interpreted as an analysis of the asymptotic behavior of the ϵ-first strategy.

In its simplest form the ϵ-greedy strategy is sub-optimal because asymptotically, the constant factor ϵ prevents the strategy from getting arbitrarily close to the optimal lever. A natural variant of the ϵ-greedy strategy is what we will call here the **ϵ-decreasing strategy**. The ϵ-decreasing strategy consists of using a decreasing ϵ for getting arbitrarily close to the optimal strategy asymptotically (the ϵ-decreasing strategy, with an ϵ function carefully chosen, achieves zero regret). The lever with the highest estimated mean is always pulled except when a random lever is pulled instead with an ϵ_t frequency where t is the index of the current round. The value of the decreasing ϵ_t is given by $\epsilon_t = \min\left\{1, \frac{\epsilon_0}{t}\right\}$ where $\epsilon_0 > 0$. The choice of ϵ_0 is left to the user. The first analysis of the ϵ-decreasing strategy seems to be by Cesa-Bianchi and Fisher [5] for an algorithm called GreedyMix. GreedyMix slightly differs from the ϵ-decreasing strategy as just presented because it uses a decreasing factor of $\log(t)/t$ instead of $1/t$. Cesa-Bianchi and Fisher prove, for specific families of reward distributions, a $\mathcal{O}(\log(T)^2)$ regret for GreedyMix where T is the number of rounds. This result is improved by Auer et al. [1] who achieve a $\mathcal{O}(\log(T))$ regret for the ϵ-decreasing strategy as presented above with some constraint over the choice of the value ϵ_0. Four other strategies are presented in [1] beside ϵ-decreasing. Those strategies are not described here because of the level of detail this would require. We chose the ϵ-decreasing strategy because the experiments by [1] seem to show that, with carefully chosen parameters, ϵ-decreasing is always as good as other strategies.

A variant of the ϵ-decreasing algorithm is introduced in [20]. The lever of highest estimated mean is always pulled except when the *least-taken* lever is pulled with a probability of $4/(4+m^2)$ where m is the number of times the least-taken lever has already been pulled. In the following, we refer to this method as the **LeastTaken** strategy. Used as such, the LeastTaken method is likely to provide very poor results in situations where the number of levers K is significant compared to the horizon H. Therefore, as for the other methods, we introduce an exploration parameter $\epsilon_0 > 0$ such that the probability of selecting the leasttaken lever is $4\epsilon_0/(4+m^2)$. The choice of ϵ_0 is left to the user. The LeastTaken method is only introduced as a heuristic (see [21]), but it is clear that this method, modified or not, is a zero-regret strategy.

2.2 The SoftMax Strategy and Probability Matching Variants

The **SoftMax strategy** consists of a random choice according to a Gibbs distribution. The lever k is chosen with probability $p_k = e^{\widehat{\mu}_k/\tau} / \sum_{i=1}^n e^{\widehat{\mu}_i/\tau}$ where $\widehat{\mu}_i$ is the estimated mean of the rewards brought by the lever i and $\tau \in \mathbb{R}^+$ is a parameter called the *temperature*. The choice of τ's value is left to the user. SoftMax appears to have been proposed first in [15]. More generally, all methods that choose levers according to a probability distribution reflecting how likely the levers are to be optimal, are called *probability matching* methods.

The SoftMax strategy (also called Boltzmann Exploration) could be modified in the same way as the ϵ-greedy strategy into **decreasing SoftMax** where the temperature decreases with the number of rounds played. The decreasing SoftMax is identical to the SoftMax but with a temperature $\tau_t = \tau_0/t$ that depends on the index t of the current round. The choice of the value of τ_0 is left to the user. The decreasing SoftMax is analyzed by Cesa-Bianchi and Fisher (1998) in [5] with the SoftMix algorithm. The SoftMix slightly differs from the decreasing SoftMax as just presented since it uses a temperature decreasing with a $\log(t)/t$ factor instead of a $1/t$ factor. The SoftMix strategy has the same guarantees than the GreedyMix strategy (see here above). To our knowledge, no result is known for the $1/t$ decreasing factor, but results similar to the ϵ-decreasing strategy are expected. The experiments in [5] show that GreedyMix outperforms SoftMix, though not significantly. Therefore, for the sake of simplicity, only the GreedyMix equivalent is used in our experiments (Section 4).

A more complicated variant of the SoftMax algorithm, the **Exp3** "exponential weight algorithm for exploration and exploitation" is introduced in [2]. The probability of choosing the lever k at the round of index t is defined by

$$p_k(t) = (1-\gamma)\frac{w_k(t)}{\sum_{j=1}^K w_j(t)} + \frac{\gamma}{K}, \tag{1}$$

where $w_j(t+1) = w_j(t) \exp\left(\gamma \frac{r_j(t)}{p_j(t)K}\right)$ if the lever j has been pulled at time t with $r_j(t)$ being the observed reward, $w_j(t+1) = w_j(t)$ otherwise. The choice of the value of the parameter $\gamma \in (0,1]$ is left to the user. The main idea is to divide the actual gain $r_j(t)$ by the probability $p_j(t)$ that the action was chosen. For a modified version of Exp3, with γ decreasing over time, it is shown by [3], that a regret of $\mathcal{O}(\sqrt{KT\log(K)})$ is achieved. The Exp3 strategy was originally proposed by Auer et al. (2002) in [3] along with five variants for the non-stochastic bandit problem. The other variants are not described here due to the level of detail required. Note also that the non-stochastic bandit is a generalization of the stochastic one with weaker assumptions, thus the theoretical guarantees of Exp3 still apply here.

More specific methods exist in the literature if additional assumptions are made about the reward distributions. We will not cover the case of boolean reward distributions (too specific for this paper, see [25] for such methods). Nevertheless, let us consider the case where Gaussian reward distributions are assumed; [25] describes a method that explicitly estimates $p_i = \mathrm{P}[\mu_i = \mu^*]$ under that assumption. This method was also previously introduced in [18] but limited to the two-armed bandit. The explicit formula would require a level of details that goes beyond the scope of this paper and will not be given here. This method will be referred to in the following as the **GaussMatch** method.

2.3 The Interval Estimation Strategy

A totally different approach to the exploration problem is to attribute to each lever an "optimistic reward estimate" within a certain confidence interval and to greedily choose the lever with the highest optimistic mean. Unobserved or infrequently observed levers will have an over-valued reward mean that will lead to further exploration of those levers. The more a lever is pulled and the closer its optimistic reward estimate will be to the true reward mean. This approach called *Interval Estimation* (referred as IntEstim in the following) is due to Kaelbling (1993) in [12]. To each lever is associated the $100 \cdot (1-\alpha)\%$ reward mean upper bound where α is a parameter in $(0,1)$ whose exact value is left to the user. At each round, the lever of highest reward mean upper bound is chosen. Note that smaller α values lead to more exploration.

In [12], the IntEstim algorithm is applied to boolean rewards. Since we are dealing here with real distributions, we will assume that the rewards are normally distributed and compute the upper bound estimate according based on that assumption. Formally, for a lever observed n times with $\widehat{\mu}$ as empirical mean and $\widehat{\sigma}$ as empirical standard deviation, the α upper bound is defined by $u_\alpha = \widehat{\mu} + \frac{\widehat{\sigma}}{\sqrt{n}}\mathrm{c}^{-1}(1-\alpha)$ where c is the cumulative normal distribution function defined by $\mathrm{c}(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} exp(-x^2/2)dx$. Choosing normal distributions is arbitrary but seems reasonable if nothing more is known about the lever reward distributions. In this paper, this choice is also motivated by the fact that part of the experiments have been performed with normally distributed levers (see Section 4).

Many variants of IntEstim have been proposed in the generalized model of MDPs. 32 different algorithms are discussed in [17] (IEQL+ may be the most well known of the introduced variant). But in the simpler stateless situation, all these variants are equivalent to IntEstim.

To our knowledge, no theoretical results are known about the IntEstim algorithm for the real-valued bandit problem (as opposed to the simpler boolean-valued bandit where the rewards could take only the values 0 and 1). In its simplest form, as just presented, IntEstim is clearly not a zero-regret strategy (it suffices to consider the case where the optimal lever has been initially very poorly estimated), but a proper control of the parameter α could make this strategy achieve zero regret.

3 The Poker Strategy

The "Price of Knowledge and Estimated Reward" (Poker) strategy relies on three main ideas: pricing uncertainty, exploiting the lever distribution, and taking into account the horizon.

The first idea is that a natural way of balancing exploration and exploitation is to assign a price to the knowledge gained while pulling a particular lever. This idea has been already used in the bandit literature. In particular, the notion of "value of information" has been intensively studied in several domains and goes far beyond the scope of this paper. In the bandit literature, it is sometimes referred to as "exploration bonuses" [17, 6]. The objective is to quantify the uncertainty in the same units as the rewards.

The second idea is that the properties of unobserved levers could potentially be estimated, to a certain extent, from the levers already observed. This is particularly useful when there are many more levers than rounds. Most of the work on the bandit problem is centered on an asymptotic viewpoint over the number of rounds, but we believe that in many practical situations, the number of rounds may be significantly smaller than the number of levers (see next section).

The third observation is that the strategy must explicitly take into account the horizon H, i.e., the number of rounds that remains to be played. Indeed, the amount of exploration clearly depends on H, e.g., for $H = 1$, the optimal strategy is reduced to pure exploitation, that is to choosing the lever with the highest estimated reward. In particular, the horizon value can be used to estimate the *price* of the knowledge acquired.

3.1 Algorithm

Let $\mu^* = \max_i\{\mu_i\}$ be the highest reward mean and let j_0 be the index of the best reward mean estimate: $j_0 = \underset{i}{\operatorname{argmax}}\{\widehat{\mu}_i\}$. We denote by $\widehat{\mu}^*$ the reward mean of j_0. By definition of μ^*, $\mu^* \geq \mu_{j_0} = \widehat{\mu}^*$. $\mu^* - \widehat{\mu}^*$ measures the reward mean improvement. We denote the expected reward improvement by $\delta_\mu = \mathrm{E}[\mu^* - \widehat{\mu}^*]$.

At each round, the expected gain when pulling lever i is given by the product of the expected reward mean improvement, δ_μ, and the probability of an improvement $\mathrm{P}[\mu_i - \widehat{\mu}^* \geq \delta_\mu]$. Over a horizon H, the knowledge gained can be exploited H times. Thus, we can view $\mathrm{P}[\mu_i \geq \widehat{\mu}^* + \delta_\mu]\delta_\mu H$ as an estimate of the knowledge acquired if lever i is pulled. This leads us to define the lever pricing formula for the POKER strategy as:

$$p_i = \widehat{\mu}_i + \mathrm{P}[\mu_i \geq \widehat{\mu}^* + \delta_\mu]\delta_\mu H, \tag{2}$$

where p_i is the price associated to the lever i by the casino (or the value of lever i for the gambler). The first term, $\widehat{\mu}_i$, is simply the estimated reward mean associated to the lever i, the second term an estimate of the knowledge acquired when lever i is pulled.

Let us also examine how the second term is effectively computed. Let $\widehat{\mu}_{i_1} \geq \cdots \geq \widehat{\mu}_{i_q}$ be the ordered estimated means of the levers already observed. We chose to define the estimated reward improvement by $\delta_\mu = (\widehat{\mu}_{i_1} - \widehat{\mu}_{i_{\sqrt{q}}})/\sqrt{q}$. The index choice $f(q) = \sqrt{q}$ is motivated by its simplicity and the fact that it ensures both $f(q) \to \infty$ (variance minimization) and $f(q)/q \to 0$ (bias minimization) when $q \to \infty$. Empirically, it has also been shown to lead to good results (see next section).

Let $\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right)$ be the normal distribution. Let $\widehat{\mu}_i$ be the mean estimate, $\widehat{\sigma}_i$ be the standard deviation estimate and n_i the number of pulls for the lever i, the probability $\mathrm{P}[\mu_i \geq \widehat{\mu}^* + \delta_\mu]$ can be approximated by

$$\Phi_{\widehat{\mu}_i, \frac{\widehat{\sigma}_i}{\sqrt{n_i}}}(\widehat{\mu}^* + \delta_\mu) = \int_{\widehat{\mu}^*+\delta_\mu}^{\infty} \mathcal{N}\left(x, \widehat{\mu}_i, \frac{\widehat{\sigma}_i}{\sqrt{n_i}}\right) dx. \tag{3}$$

This would be the exact probability if $\widehat{\mu}_i$ followed a normal distribution. Note that the central limit theorem guarantees that, in the limit, the mean estimate $\widehat{\mu}_i$ of the reward distribution is normally distributed.

Algorithm 1 shows the pseudocode of the procedure POKER which takes three arguments: the reward function $\mathbf{r} : [1, K] \to \mathbb{R}$, the number of levers $K \in \mathbb{N}^*$ and the number of rounds to be played $T \in \mathbb{N}^*$. In the pseudocode, $n[i]$ represents the number of times lever i has been pulled. $\mu[i]$ (resp. $\sigma[i]$), the reward mean (resp. the estimate of the reward standard deviation) of the lever i, is used as a shortcut for $\frac{r[i]}{n[i]}$ (resp $\sqrt{\frac{r_2[i]}{n[i]} - \frac{r[i]^2}{n[i]^2}}$). $\widehat{\mathrm{E}}_{k,n[k]>0}$ denotes the empirical mean taken over the set of levers previously pulled.

A round is played at each iteration through the loop of lines $2 - 14$. The computation of the price for each lever is done at lines $7 - 12$. The estimates of the mean and standard deviation of each lever are computed in lines $8 - 9$. Note that if the lever has not been observed yet, then the set of levers already observed is used to provide *a priori* estimates. The price is computed at line 10. The *initialization* of the algorithm has been omitted to improve readability. The initialization simply consists of pulling twice two random levers so that i_0 and i_1 are well-defined at line 4.

Algorithm 1. *Poker*(r, K, T)

```
 1: for $i = 0$ to $K$ do $n[i] \leftarrow r[i] \leftarrow r_2[i] \leftarrow 0$ end for
 2: for $t = 1$ to $T$ do
 3:    $q \leftarrow |\{i, r[i] > 0\}|$
 4:    $i_0 \leftarrow \operatorname{argmax}_i\{\mu[i]\}$ ; $i_1 \leftarrow j$ such that $|\{i, \mu[i] > \mu[j]\}| = \sqrt{q}$
 5:    $\delta_\mu \leftarrow (\mu[i_0] - \mu[i_1]) / \sqrt{q}$ ; $\mu^* \leftarrow \operatorname{argmax}_i\{\mu[i]\}$
 6:    $p_{max} \leftarrow -\infty$ ; $i_{max} \leftarrow$ UNDEFINED
 7:    for $i = 1$ to $K$ do
 8:       if $n[i] > 0$ then $\mu \leftarrow \mu[i]$ else $\mu \leftarrow \widehat{\mathrm{E}}_{k,n[k]>0}[\mu[k]]$ endif
 9:       if $n[i] > 1$ then $\sigma \leftarrow \sigma[i]$ else $\sigma \leftarrow \widehat{\mathrm{E}}_{k,n[k]>1}[\sigma[k]]$ endif
10:       $p \leftarrow \mu + \delta_\mu (T - t) \int_{\mu^*+\delta_\mu}^{\infty} \mathcal{N}\left(x, \mu[i], \frac{\sigma[i]}{\sqrt{n[i]}}\right) dx$
11:       if $p > p_{max}$ then $p_{max} \leftarrow p$, $i_{max} \leftarrow i$ endif
12:    end for
13:    $r \leftarrow r(i_{max})$ ; $n[i_{max}] += 1$ ; $r[i_{max}] += r$ ; $r_2[i_{max}] += r^2$
14: end for
```

Algorithm 1 gives an *offline* presentation of POKER, but POKER is in fact intrinsically an *online* algorithm. The horizon value $T - t$ (line 10 in Algorithm 1) could simply be set to a constant value. Notice that the amount of exploration has to be controlled in some way. Most of the algorithms presented in section 2 have an exploration tuning parameter. We believe that the horizon is an intuitive and practical exploration control parameter, especially compared to the τ parameter for the SOFTMAX or the α parameter of INTESTIM.

It is easy to see that POKER is a zero-regret strategy. The proof is very technical however and requires more space than we can afford here. The following gives a sketch of the proof.

3.2 POKER is a Zero-Regret Strategy - Sketch of the Proof

Let us consider a game played by POKER where rounds are indexed by t such as $t = 1$ refers to the first round and $t = H$ refers to the last round. The proof has two parts: first, an argument showing that all levers are pulled a significant number of times; then, using the first part, establishing the fact that a "bad" lever cannot be pulled too frequently.

Let $m_i(t)$ be the number of times the lever i has been pulled till round t and assume that all rewards are bounded by $R > 0$. Then, by Hoeffding's inequality, $\mathrm{P}\left[\mu_i \geq \widehat{\mu}_i + \delta_\mu\right] \leq \exp(-2m_i(t)\frac{\delta_\mu^2}{R^2})$ for $i = 1, \ldots, K$. Since $\widehat{\mu}^* > \widehat{\mu}_i$, this implies that: $\mathrm{P}\left[\mu_i \geq \widehat{\mu}^* + \delta_\mu\right] \leq \exp(-2m_i(t)\frac{\delta_\mu^2}{R^2})$ for $i = 1, \ldots, K$.

Now, it is clear that $m_i(H)$ tends to infinity on average when H tends to infinity. Just consider that $p_i(t)$ at fixed t tends to infinity when H tends to infinity. The same argument shows also that for any $\epsilon > 0$, $m_i(\epsilon H)$ tends to infinity when H tends to infinity.

Let m_H be such that $\exp(-2m_H\frac{\delta_\mu^2}{R^2})\delta_\mu H < r/2$. Given the asymptotic behavior of m_i just discussed, there exists t_1 such that for all i, $m_i(t_1) > m_H$ with probability q. Let $r > 0$ be a given regret. Assume that for a given lever distribution, playing POKER implies that there exists a lever i and a constant $\alpha > 0$ such that $m_i(H) > \alpha H$ (frequent lever assumption) and $\mu_i < \mu^* - r$ (poor lever assumption) for any H. Let i be such a lever. The existence of i is the negation of the zero-regret property. Choose H large enough such that $\frac{t_1}{H} < \alpha$.

The probability that the lever i is played at least once in the interval is $[t_1, H]$ is expressed by the probability that the price p_i be the highest price, formally $\mathrm{P}[\exists t \geq t_1 : p_i(t) \geq p^*(t)]$. The inequality $\exp(-2m_H\frac{\delta_\mu^2}{R^2})\delta_\mu H < r/2$ implies that (the quantifier and argument t are omitted for simplicity):

$$\mathrm{P}[p_i \geq p^*] \leq \mathrm{P}\left[\widehat{\mu}_i + \frac{r}{2} - \widehat{\mu}^* > 0\right]. \tag{4}$$

Since all levers have already been pulled at least m_H times by definition of t_1, by Hoeffding's inequality (using the fact that $\mu_i + \frac{r}{2} - \mu^* < -\frac{r}{2}$) the probability of that event is bounded as follows:

$$\mathrm{P}\left[\widehat{\mu}_i + \frac{r}{2} - \widehat{\mu}^* > 0\right] \leq \mathrm{P}\left[\widehat{\mu}_i - \widehat{\mu}^* > \mu_i - \mu^* + \frac{r}{2}\right] \leq \exp[-m_H\frac{r^2}{2R^2}]. \tag{5}$$

Thus, the lever i has a probability greater than q of not verifying $m_i(H) > \alpha H$ for H large enough. Additionally, by choosing H large enough, the probability q can be made arbitrarily close to 1. This conclusion contradicts the uniform existence (for any H) of the lever i. POKER is a zero-regret strategy.

4 Experiments

This section describes our experiments for evaluating several strategies for the bandit problem using two datasets: an artificially generated dataset with known and controlled distributions and a real networking dataset.

Many bandit methods requires all levers to be pulled once (resp. twice) before the method actually begins in order to obtain an initial mean (resp. a variance) estimate. In particular, IntEstim requires two pulls per lever, see [17]. However this *pull-all-first* initialization is inefficient when a large number of levers is available because it does not exploit the information provided by the known lever distribution (as discussed in the second idea of Poker here above). Therefore, in our experiments, the mean and variance of unknown levers, whenever required, are estimated thanks to the known lever distribution. In order to obtain a fair comparison, the formula in use is always identical to the formula used in Poker.

4.1 Randomly Generated Levers

The first dataset is mainly motivated by its simplicity. Since normal distributions are perhaps the most simple non-trivial real distributions, we have chosen to generate normally distributed rewards. This choice also fits the underlying assumptions for the algorithms IntEstim and GaussMatch.

The dataset has been generated as follows: all levers are normally distributed, the means and the standard deviations are drawn uniformly from the open interval $(0, 1)$. The objective of the agent is *to maximize the sum of the rewards*. The dataset was generated with 1000 levers and 10 000 rounds. The bandit strategies have been tested in three configurations: 100 rounds, 1000 rounds, 10 000 rounds which correspond to the cases of less rounds than levers, as many rounds as levers, or more rounds than levers. Although we realize that most of the algorithms we presented were designed for the case where the number of rounds is large compared to the number of lever, we believe (see here below or [4]) that the configuration with more levers than rounds is in fact an important case in practice. Table 1 (columns R-100, R-1k and R-10k) shows the results of our experiments obtained with 10 000 simulations. Note that the numbers following the name of the strategies correspond to the tuning parameter values as discussed in section 2.

4.2 URLs Retrieval Latency

The second dataset corresponds to a real-world data retrieval problem where redundant sources are available. This problem is also commonly known as the *Content Distribution Network* problem (CDN) (see [13] for a more extensive introduction). An agent must retrieve data through a network with several redundant sources available. For each retrieval, the agent selects one source and waits until the data is retrieved[2]. The objective of the agent is *to minimize the sum of the delays* for the successive retrievals.

In order to simulate the retrieval latency problem under reproducible conditions, we have used the home pages of more than 700 universities as sources. The home pages have been retrieved roughly every 10 min for about 10 days (∼1300 rounds), the retrieval latency being recorded each time in milliseconds[3]. Intuitively each page is associated to a lever, and each latency is associated to a

[2] We assume that the agent could try only *one* source at a time, in practice he will only be able to probe simultaneously a very limited number of sources.

[3] The dataset has been published under a public domain license, making it accessible for further experiments in the same conditions. It can be accessed from `sourceforge.net/projects/bandit`.

Table 1. Experimental results for several bandit algorithms. The strategies are compared in the case of several datasets. The *R-x* datasets corresponds to a maximization task with random Gaussian levers (the higher the score, the better). The *N-x* datasets corresponds to a minimization task with levers representing retrieval latencies (the lower the score, the better). The numbers following the strategy names are the tuning parameters used in the experiments.

Strategies	**R-100**	**R-1k**	**R-10k**	**N-130**	**N-1.3k**
Poker	0.787	0.885	0.942	203	132
ϵ-greedy, 0.05	0.712	0.855	0.936	733	431
ϵ-greedy, 0.10	0.740	0.858	0.916	731	453
ϵ-greedy, 0.15	0.746	0.842	0.891	715	474
ϵ-first, 0.05	0.732	0.906	0.951	735	414
ϵ-first, 0.10	0.802	0.893	0.926	733	421
ϵ-first, 0.15	0.809	0.869	0.901	725	411
ϵ-decreasing, 1.0	0.755	0.805	0.851	738	411
ϵ-decreasing, 5.0	0.785	0.895	0.934	715	413
ϵ-decreasing, 10.0	0.736	0.901	0.949	733	417
LeastTaken, 0.05	0.750	0.782	0.932	747	420
LeastTaken, 0.1	0.750	0.791	0.912	738	432
LeastTaken, 0.15	0.757	0.784	0.892	734	441
SoftMax, 0.05	0.747	0.801	0.855	728	410
SoftMax, 0.10	0.791	0.853	0.887	729	409
SoftMax, 0.15	0.691	0.761	0.821	727	410
Exp3, 0.2	0.506	0.501	0.566	726	541
Exp3, 0.3	0.506	0.504	0.585	725	570
Exp3, 0.4	0.506	0.506	0.594	728	599
GaussMatch	0.559	0.618	0.750	327	194
IntEstim, 0.01	0.725	0.806	0.844	305	200
IntEstim, 0.05	0.736	0.814	0.851	287	189
IntEstim, 0.10	0.734	0.791	0.814	276	190

(negative) reward. The bandit strategies have been tested in two configurations: 130 rounds and 1300 rounds (corresponding respectively to $1/10^{th}$ of the dataset and to the full dataset). Table 1 (columns N-130 and N-1.3k) shows the results which correspond to the average retrieval latencies per round in milliseconds. The results have been obtained through 10 000 simulations (ensuring that the presented numbers are significant). The order of the latencies was randomized through a random permutation for each simulation.

4.3 Analysis of the Experimental Results

Let us first examine the ϵ-greedy strategy and its variants. Note that all ϵ-greedy variants have similar results for carefully chosen parameters. In particular, making the ϵ decrease does not significantly improve the performance. The ϵ_0 (the real parameter of the ϵ-decreasing strategy) also seems to be less intuitive than the ϵ parameter of the ϵ-greedy strategy. Although very different from the ϵ-greedy, the SoftMax strategy leads to very similar results. But its Exp3 variant seems to have a rather poor performance, its results are worse than any other strategy independently of the parameters chosen. The reason probably lies in the fact that the Exp3 has been designed to optimize its asymptotic behavior which does not match the experiments presented here.

The two "pricing" strategies Poker and IntEstim significantly outperform all of the other strategies on the networking dataset, by a factor of 2 for IntEstim and a factor of 3 for Poker. Against the random generated dataset, IntEstim performs significantly worse than the other strategies, a rather unexpected result since the generated dataset perfectly fits the IntEstim assumptions, while Poker is always as good as the best strategy for any parameter. We do not have yet proofs to justify the "good" behavior of the two pricing methods on the networking dataset, but this seems related to the "shape" of the networking data. The networking data proves to be very peaky with latencies that cover a wide range of values from 10 ms to 1000 ms with peaks to 10000 ms. With that data, exploration needs to be carefully handled because trying a new lever could prove to be both a major improvement or a major cost. It seems that strategies with a dynamic approach for the level of exploration achieve better results than those where the amount of exploration is fixed *a priori*.

5 Conclusion

In the case where the lever reward distributions are normally distributed, simple strategies with no particular theoretical guarantees such as ϵ-greedy tend to be hard to beat and significantly outperform more complicated strategies such as Exp3 or *Interval Estimation*. But, the ranking of the strategies changes significantly when switching to real-world data. Pricing methods such as *Interval Estimation* or Poker significantly outperform naive strategies in the case of the networking data we examined. This empirical behavior was rather unexpected since the strategies with the best asymptotic guarantees do not provide the better results, and could not have been inferred from a simple comparison of the theoretical results known so far. Since this is, to our knowledge, the first attempt to provide a common evaluation of the most studied bandit strategies, the comparison should still be viewed as preliminary. Further experiments with data from different tasks might lead to other interesting observations. We have made the experimental data we used publicly available and hope to collect, with the help of other researchers, other datasets useful for benchmarking the bandit problem that could be made available from the same web site.

References

1. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite Time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2/3):235–256, 2002.
2. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a Rigged Casino: the Adversarial Multi-Armed Bandit Problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pages 322–331. IEEE Computer Society Press, Los Alamitos, CA, 1995.
3. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
4. B. Awerbuch and R. Kleinberg. Adaptive Routing with End-to-End feedback: Distributed Learning and Geometric Approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC 2004)*, pages 45–53, 2004.

5. N. Cesa-Bianchi and P. Fischer. Finite-Time Regret Bounds for the Multiarmed Bandit Problem. In *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, pages 100–108. Morgan Kaufmann, San Francisco, CA, 1998.
6. R. W. Dearden. *Learning and Planning in Structured Worlds.* PhD thesis, University of British Columbia, 2000.
7. E. Even-Dar, S. Mannor, and Y. Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In *Fifteenth Annual Conference on Computational Learning Theory (COLT)*, pages 255–270, 2002.
8. E. Frostig and G. Weiss. Four proofs of gittins' multiarmed bandit theorem. *Applied Probability Trust*, 1999.
9. J. C. Gittins. *Multiarmed Bandits Allocation Indices.* Wiley, New York, 1989.
10. J. C. Gittins and D. M. Jones. A dynamic allocation indices for the sequential design of experiments. In *Progress in Statistics, European Meeting of Statisticians*, volume 1, pages 241–266, 1974.
11. J. P. Hardwick and Q. F. Stout. Bandit Strategies for Ethical Sequential Allocation. *Computing Science and Statistics*, 23:421–424, 1991.
12. L. P. Kaelbling. *Learning in Embedded Systems.* MIT Press, 1993.
13. B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. In *SIGCOMM IMW*, pages 169–182, November 2001.
14. N. Littlestone and M. K. Warmuth. The Weighted Majority Algorithm. In *IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989.
15. D. Luce. *Individual Choice Behavior.* Wiley, 1959.
16. S. Mannor and J. N. Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. In *Sixteenth Annual Conference on Computational Learning Theory (COLT)*, 2003.
17. N. Meuleau and P. Bourgine. Exploration of Multi-State Environments: Local Measures and Back-Propagation of Uncertainty. *Machine Learning*, 35(2):117–154, 1999.
18. R. L. Rivest and Y. Yin. Simulation Results for a New Two-Armed Bandit Heuristic. Technical report, Laboratory for Computer Science, M.I.T., February 1993.
19. H. Robbins. Some Aspects of the Sequential Design of Experiments. In *Bulletin of the American Mathematical Society*, volume 55, pages 527–535, 1952.
20. M. Strens. *Learning, Cooperation and Feedback in Pattern Recognition.* PhD thesis, Physics Department, King's College London, 1999.
21. M. Strens. A Bayesian Framework for Reinforcement Learning. In *Proceedings of the 7th International Conf. on Machine Learning*, 2000.
22. R. S. Sutton. Integrated Architecture for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In *Proceedings of the seventh international conference (1990) on Machine learning*, pages 216–224. Morgan Kaufmann Publishers Inc., 1990.
23. P. Varaiya, J. Walrand, and C. Buyukkoc. Extensions of the multiarmed bandit problem: The discounted case. In *IEEE Transactions on Automatic Control*, volume AC-30, pages 426–439, 1985.
24. C. J. C. H. Watkins. *Learning from Delayed Rewards. Ph.D. thesis.* Cambridge University, 1989.
25. J. Wyatt. *Exploration and Inference in Learning from Reinforcement.* PhD thesis, University of Edinburgh, 1997.