Sub-sampling for Multi-armed Bandits

Akram Baransi[1], Odalric-Ambrym Maillard[1], and Shie Mannor[1]

Department of Electrical Engineering,
Technion - Israel Institute of Technology, Haifa, Israel
abaransi@tx.technion.ac.il,
odalric-ambrym.maillard@ens-cachan.org, shie@ee.technion.ac.il

Abstract. The stochastic multi-armed bandit problem is a popular model of the exploration/exploitation trade-off in sequential decision problems. We introduce a novel algorithm that is based on sub-sampling. Despite its simplicity, we show that the algorithm demonstrates excellent empirical performances against state-of-the-art algorithms, including Thompson sampling and KL-UCB. The algorithm is very flexible, it does need to know a set of reward distributions in advance nor the range of the rewards. It is not restricted to Bernoulli distributions and is also invariant under rescaling of the rewards. We provide a detailed experimental study comparing the algorithm to the state of the art, the main intuition that explains the striking results, and conclude with a finite-time regret analysis for this algorithm in the simplified two-arm bandit setting.

Keywords: Multi-armed Bandits, Sub-sampling, Reinforcement Learning.

1 Introduction

In sequential decision making under uncertainty, the main dilemma that a decision maker faces is to explore, or not to explore. One of these problems is the popular stochastic multi-armed bandit problem, termed in reference to the 19th century gambling game and introduced by [23, 20]. It illustrates the fundamental trade-off between *exploration*, that is, making decisions that improve the knowledge of the environment, and *exploitation*, that is, choosing the decision that has maximized the previous payoff. Classically, each decision is referred to as an "arm". There is a finite set of arms and each arm, when pulled, returns a real value, called the *reward*, which is independently and identically drawn from an unknown distribution. At each time step the decision maker chooses an arm based on the sequence of rewards that has been observed so far, pulls this arm and observes a new sample from the corresponding unknown underlying distribution. The objective is to find a policy for choosing the next arm to be pulled, that maximizes the sum of the expected rewards, or equivalently minimize the expected *regret*, that is the loss caused by not pulling the best arm at each time step. If $\mathcal{A}$ denotes the set of arms and $\{\mu_a\}_{a\in\mathcal{A}}$ the mean reward of the distribution of each arm, we denote $\star \in \operatorname{argmax}_{a\in\mathcal{A}} \mu_a$ an optimal arm and the (expected) regret of an algorithm that pulls arms $a_1, \ldots, a_T$ up to time T is classically defined as

$$\mathfrak{R}_t = \mathbb{E}\left[\sum_{t=1}^{T}(\mu_\star - \mu_{a_t})\right]. \tag{1}$$

T. Calders et al. (Eds.): ECML PKDD 2014, Part I, LNCS 8724, pp. 115–131, 2014.

Previous Work. Since the formulation of the problem by Robbins (1952), the regret, that measures the cumulative loss resulting from pulling sub-optimal arms, has been a popular criterion for assessing the quality of a strategy. Gittins index based policies ([13, 14, 12]), which were initially introduced by Gittins in 1979, is a family of Bayesian-optimal policies that based on indices that fully characterize each arm given the current history of the game, and at each time step the arm with the highest index will be pulled. However, the high computational cost of Gittins indices and the fact that they are practically limited to a specific set of distributions, arose the need of modifying the policies and make them more efficient. In [8], extending the seminal work of [18], the authors characterized the achievable performance. They showed that under suitable conditions on the possible distributions associated to each arm, any policy that is "admissible" (that is, not grossly under-performing, see [18] for details) must satisfy the following asymptotic lower-performance bound

$$\liminf_{T\to\infty} \frac{\mathfrak{R}_T}{\log(T)} \geqslant \sum_{a:\mu_a<\mu_\star} \frac{\mu_\star - \mu_a}{\mathcal{K}_{\inf}\left(\nu_a;\mu_\star\right)}\,, \tag{2}$$

where $\mathcal{K}_{\inf}\left(\nu_a;\mu_\star\right)$ is an information-theoretic quantity which measures the minimal Kullback-Leibler divergence between ν_a and distributions in the model that have expectations larger than $\mu_\star$. In the same papers, [18], [10], [8] suggested that Gittins indices can be approximated by quantities that can be interpreted as upper bounds of confidence intervals.

In [1], the generic class of index policies termed **`UCB`** (**U**pper **C**onfidence **B**ounds) was introduced, together with an asymptotic analysis of their performance. [5] provided the first finite time analysis for a particular variant of **`UCB`** based on Hoeffding's inequality, showing that the regret grows logarithmically with the time horizon T. A few algorithms from the **`UCB`** family have been recently introduced such as **`UCB`**-V ([4]), **`MOSS`** ([3]), Improved-**`UCB`** ([6]), as well as the recent Kullback-Leibler-based algorithms **`DMED`** ([15]), **`Kinf`** ([19]), **`kl-UCB`** ([11]) and **`KL-UCB`** ([9]), that were shown to be first-order optimal.

Besides Gittins index and the **`UCB`**-type algorithms, another important class of algorithms is that introduced by Thompson ([23, 24]), and called **`Thompson sampling`**. The algorithm assumes that the arms' distributions belong to a parametric family of distributions $\mathcal{P} = \{p(.|\theta), \theta \in \Theta\}$ where $\Theta \subseteq R$, it starts by putting a prior distributions on each one of the arms' parameters, and at each time step a posterior distribution is maintained according to the rewards observed so far. In practice each different $\mathcal{P}$ leads to a different implementation of the algorithm. At each time step, this Bayesian algorithm draws one sample for each arm from its posterior, then pulls the arm that maximizes the expected reward given that parameter. Recently, the analysis developed in [9] enabled to tackle the first frequentist optimal bound for the Thompson-sampling algorithm ([16]) in case of a family of Bernoulli distributions, thus proving that this algorithm also achieves optimality with a the regret that grows logarithmically with T. See also [2], as well as the recent extension to another class of distributions in [17].

Contribution. In this paper we introduce a novel algorithm called **`BESA`** (**B**est **E**mpirical **S**ampled **A**verage) for the stochastic multi-armed bandit problem (see Section 2). The

Algorithm 1. **BESA**(a,b) for a two-arm bandit

Require: Two arms a, b, current time t.

1. Sample $I_t^a \sim \texttt{Wr}(N_t(a); N_t(b))$ and $I_t^b \sim \texttt{Wr}(N_t(b); N_t(a))$.
2. Define $\tilde{\mu}_{t,a} = \widehat{\mu}(X^a_{1:N_t(a)}(I_t^a))$ and $\tilde{\mu}_{t,b} = \widehat{\mu}(X^b_{1:N_t(b)}(I_t^b))$.
3. Choose (break ties by choosing the arm with the smaller N_t)
$$a_t = \operatorname*{argmax}_{a' \in \{a,b\}} \tilde{\mu}_{t,a'} .$$

algorithm has a different flavor than previously introduced algorithms. It is *not* based on the computation of an empirical confidence bounds but rather on the sampling of some specific quantity. It is for that reason related in spirit to the Thompson sampling algorithm. However, unlike Thompson sampling, **BESA** does not rely on a parametric set of distribution (or a prior) and is instead fully *non-parametric*. In Section 3, we compare the performance of the algorithm against state-of-the-art algorithms, including Thompson sampling and KL-UCB, in several scenarios with different types of reward distributions and show that the algorithm demonstrates excellent empirical performances against them. In Section 4, we provide a possible explanation for the strong performance of **BESA**, and then discuss its properties; Perhaps the most important property of **BESA** is its flexibility, since the same implementation can be used for any type of reward distributions, contrary to Thompson sampling or KL-UCB whose implementations differ according to the considered set of distribution. Finally in Section 5, we provide a finite-time regret bound for this algorithm in the two-arm bandit problem. We show with a rough analysis that the expected regret of the algorithm in this case is $O(\log(T))$ where T is the time horizon. The focus of the paper is to introduce and report the striking empirical performance of this *simple* and *flexible* algorithm.

Setup and Notations. We consider a multi-armed bandit setting with finitely many arms $\mathcal{A}$ and respective reward distributions $\{\nu_a\}_{a\in\mathcal{A}}$, where $\nu_a \in \mathcal{P}([0,1])$ and $\mathcal{P}([0,1])$ denotes the set of probability measures with support in $[0,1]$. We denote $\mu_a \in [0,1]$ the mean of the distribution ν_a, and $X^a_{1:n} = (X^a_1, \dots, X^a_n)$ a sample of size n, i.i.d. from ν_a. In the sequel, we use the short-hand notation $[n]$ for the set of integer $\{1, \dots, n\}$. For a set of indices $I \subset [n]$ of size m, say $I = (i_1, \dots, i_m)$, we write $X^a_{1:n}(I) = (X^a_{i_1}, \dots, X^a_{i_m})$ for the corresponding sub-sampled set. A sample of size m drawn *without replacement* from the set $[n]$ is written $I(n;m) \sim \texttt{Wr}(n;m)$. Here $\texttt{Wr}(n;m)$ denotes a distribution over sets of integers (with the convention that $\texttt{Wr}(n;m) = \delta_{[n]}$ if $m > n$, where δ refers to a Dirac distribution). Finally, for a sample S of real values, we denote $\widehat{\mu}(S)$ the average of the sample components. For instance we have $\widehat{\mu}(X^a_{1:n}) = \frac{1}{n}\sum_{i=1}^n X^a_i$. Let $\star \in \mathcal{A}$ denote an arm with maximal mean $\mu_\star$. The regret of an algorithm that pulls arms $\{a_t\}_{t\in[T]}$ up to time T is defined by (1), where the expectation is taken with respect to all sources of randomness. We also denote the number of pulls of an arm $a \in \mathcal{A}$ up to time t by $N_t(a) = \sum_{t'=1}^{t-1} \mathbb{I}\{a_{t'} = a\}$.

2 The BESA Algorithm

The main contribution of this paper is to introduce a novel algorithm, called **BESA** (**B**est **E**mpirical **S**ampled **A**verage) that uses a sub-sampling procedure in order to compare

Algorithm 2. **BESA**($\mathcal{A}$) for a multi-armed bandit

Require: Set of arms $\mathcal{A}$ of size A, current time t.
1. **if** $\mathcal{A} = \{a\}$ **then**
2. Choose $a_t = a$.
3. **else**
4. Choose $a_t = \mathbf{BESA}_t(\mathbf{BESA}_t(\{a_i\}_{1 \leqslant i < \lceil A/2 \rceil}), \mathbf{BESA}_t(\{a_i\}_{\lfloor A/2 \rfloor < i \leqslant A}))$
5. **end if**

between the empirical values of two arms. The pseudo-code of the algorithm, for two arms is provided in Algorithm 1. The version for the more general multi-armed bandit uses a tournament strategy described in Algorithm 2.

The main idea of the algorithm is to make a fair comparison between the arms: Given two arms a and b that has been pulled $n_a = N_t(a)$ and $n_b = N_t(b) > n_a$ times respectively at time t, comparing the empirical averages of the arms is not a fair comparison since a has not gotten the same number of opportunities as b to show its abilities. **BESA** compensates for this situation by sub-sampling uniformly n_a rewards out of the n_b rewards of arm b. **BESA** then compares the empirical average of the rewards from a, to the empirical average of the rewards sub-sampled from b. It finally chooses b if its computed value is larger than the one of a. We provide in Algorithm 1 a more formal and unified presentation of this strategy. If $n_b > n_a$, the sampled set (line 2) is $I_t^b \sim \mathtt{Wr}(N_t(b); N_t(a))$ (indeed $I_t^a \sim \mathtt{Wr}(N_t(a); N_t(b))$ is the full set $[N_t(a)]$ in this case). Then (line 3,4) the compared values become $\widehat{\mu}(X^a_{1:N_t(a)})$ and $\widehat{\mu}(X^b_{1:N_t(b)}(I_t^b))$.

BESA, FTL and Thompson Sampling. At first sight, **BESA** seems close to a version of the standard **F**ollow **T**he **L**eader (FTL) algorithm. This algorithm selects $\mathrm{argmax}_{a \in \mathcal{A}} \widehat{\mu}(X^a_{1:N_t(a)})$, that is the best empirical arm (with no sub-sampling). **FTL** is known to be a *bad* strategy in the bandit setting as it can lead to a linear regret in a number of situations. It is thus *a priori* striking that **BESA** can be any reasonable. On the other hand, **BESA** uses a sampling strategy in order to select the subset used to compute the sub-sampled mean. This is in this respect related in spirit to the **Thompson sampling** strategy, that is known to be both Bayesian optimal, and frequentist optimal, achieving the state-of-the-art for the bandit setting with Bernoulli distribution of rewards ([16]), or more recently distributions in the one-dimension exponential family ([17]). Note that **Thompson sampling** actually refers to a collection of algorithms whose implementation depend on the prior we have on reward distributions. Thus **Thompson sampling** for Bernoulli distributions is for instance different than **Thompson sampling** for exponential distributions. In contrast, **BESA** keeps the same form regardless of the distribution on rewards.

A Tournament for Many Arms. We extend Algorithm 1 written for two arms to the more general case of a finite set $\mathcal{A}$ of arms by using a divide-and-conquer style algorithm (see Algorithm 2). This intuitively corresponds to organizing a tournament between arms. To avoid relying too much on a specific ordering of the arms that may bias the final result (and look arbitrary), we randomly shuffle the set of arms before each decision. That is, at time t, we create a copy $\tilde{\mathcal{A}}_t$ of $\mathcal{A}$ that is obtained by shuffling $\mathcal{A}$ uniformly at random, and then output the arm **BESA**($\tilde{\mathcal{A}}_t$).

3 Numerical Experiments

Our findings show that, surprisingly, **BESA** is a strong competitor of the state-of-the-art strategies from the bandit literature. Before providing one possible explanation for this striking performance, we now report these intriguing results more precisely.

In this section, the **BESA** algorithm is compared against well-known optimal algorithms such as **KL-UCB**, **KL-UCB+** ([11, 19, 9]), **Thompson sampling** ([23, 24]) with prior Beta(1,1), **KL-UCB**-exp ([9]) and **UCB**-tuned ([5]), on different scenarios with different set of arms. To avoid implementation bias, we use the open-source code available on-line at `http://mloss.org/software/view/415` for the implementation of these algorithms. Note that these algorithms do not need parameter tuning. In each one of the scenarios detailed below, the time horizon is set to $T = 20,000$, and the scenario is run on 50,000 independent experiments. In each run, so as to have a fair comparison, the rewards of the arms are all drawn in advance, and all the algorithms are run on the same set of drawn rewards. In other words, on each of the $50,000$ runs, $\forall n \in \{1, \ldots, 20000\}, a \in \mathcal{A}$ all the algorithms will observe the same reward on the n^{th} pull of arm a. This enables us to measure the percentage of runs on which one algorithm is better than a reference one, thus providing another measure of performance, besides the empirical average cumulative. We systematically report below in Table $1, \ldots, 7$ this percentage using **BESA** as a reference. In Figures $1, \ldots, 7$, the dark gray represents the plot quartiles, while the light gray represents the upper 5 percents quantile. Finally, in section 5, we introduce the so-called balance function $\alpha_{1/2}$ (see definition 1) that acts as a complexity parameter. For clarity, we report the scaling of this function in most of the following scenarios as well.

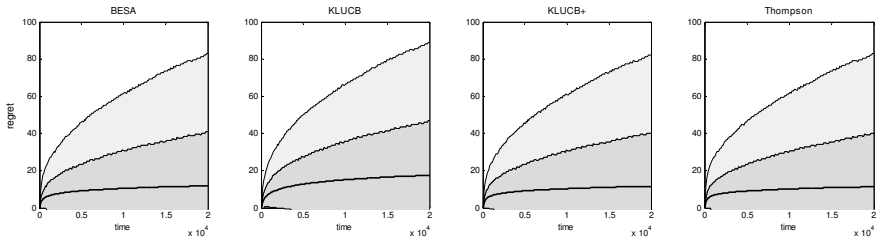

Fig. 1. Regret against time for the two-arm scenario, with $\mu_\star = 0.9$ and $\mu_a = 0.8$

Table 1. Performance measures for $T = 20,000$ in the two-arm scenario, with $\mu_\star = 0.9$ and $\mu_a = 0.8$. Complexity $\alpha_{1/2}(M, 1) = O(0.9^M)$.

	BESA	**KL-UCB**	**KL-UCB+**	**Thompson sampling**
Average regret at T	11.83	17.48	11.54	11.3
Beat **BESA**	$--$	1.82%	41.6%	58.28%
Average running time	2.86X	2.7X	3.12X	X

3.1 Scenario 1: Two Bernoulli Arms

In this scenario we consider the case of two Bernoulli arms $\mathcal{A} = \{\star, a\}$, with expectations $\mu_\star = 0.9$ and $\mu_a = 0.8$, respectively. The empirical average cumulative regret of each algorithm is shown in the first raw of Table 1, while the second raw shows the percentage of the runs on which the algorithm gave a lower regret than **BESA**, and the third shows the average run time where X denotes the average run time of the fastest algorithm. In Figure 1 the average regret is shown as a function of time. The same scenario has been considered in [11]. On one hand, from figure 1 in [11] one can conclude that the average cumulative regret of **UCB**-V is larger than 50, while all the other algorithms but **KL-UCB** have average cumulative regret between 21 and 36. On the other hand, as reported in Table 1, the average cumulative regret of **BESA** is 11.38. Thus **BESA** outperforms all the algorithms considered in [11] such as e.g. **UCB**-tuned, **DMED**, MOSS on this scenario, including **KL-UCB**. Note that **KL-UCB**+ does get a slightly lower expected regret, but does not beat **BESA** more than 50 per cent of the time. **Thompson sampling** here slightly outperforms **BESA**.

3.2 Scenario 2: Bernoulli with a Small Δ

This scenario is similar to scenario 1 but with a smaller gap Δ: We consider the case of two Bernoulli arms, with expectations $\mu_1 = 0.81$ and $\mu_2 = 0.8$ respectively. Similarly to scenario 1 the average regret, the percentage of experiments on which **BESA** was beaten and the average run time are shown in Table 2, and the cumulative regret as a function of time is shown in Figure 2. Note that the average regret of **BESA** is close to that of **KL-UCB**+ and smaller than that of **KL-UCB** and **Thompson sampling**. Interestingly enough, in addition the percentage of runs on which **BESA** is beaten by any of the state-of-the-art algorithms is smaller than 37%.

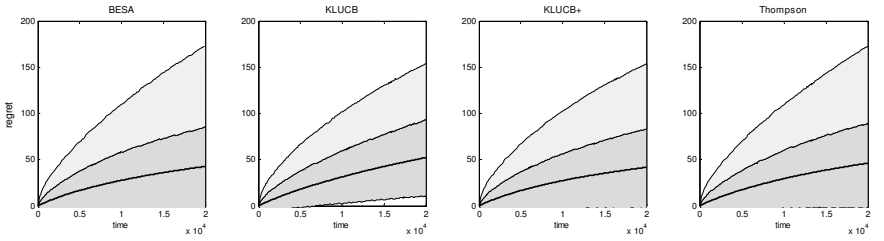

Fig. 2. Regret against time for the two-arm scenario, with $\mu_\star = 0.81$ and $\mu_a = 0.8$.

Table 2. Performance measures for $T = 20,000$ for the two-arm scenario, with $\mu_\star = 0.81$ and $\mu_a = 0.8$. Complexity $\alpha_{1/2}(M, 1) = O(0.9^M)$.

	BESA	**KL-UCB**	**KL-UCB+**	**Thompson sampling**
Average regret at T	42.6	52.34	41.71	46.14
Beat **BESA**	$--$	25.61%	36.86%	35.2%
Average running time	4.56X	2.78X	3.47X	X

3.3 Scenario 3: Bernoulli with Low Means

In this scenario we consider the scenario used in [11], and inspired by a situation, frequent in applications like marketing or Internet advertising, where the mean reward of each arm is very low. More precisely we consider a harder case which has ten Bernoulli arms, the best arm has expectation 0.1, three arms have expectation 0.05, three arms expectation 0.02, and the rest with expectation 0.01. Table 3 summarizes the results of this experiment, and the regret as a function of time is shown in Figure 3. As can be seen from Table 3 the average regret of **BESA** is much smaller than **KL-UCB** and **Thompson sampling** regrets, and it is beaten by **KL-UCB** only in 1.57% of the runs and by **Thompson sampling** only in 3.09% of the runs. It is also beaten by **KL-UCB+** less than 36% of the time. As can be seen in figure 2 of [11] the regrets of all the algorithms but **DMED+** and **KL-UCB+**, which include e.g. CP-**UCB**, **DMED**, **UCB**-Tuned, are between 100 and 400. Thus we can conclude that **BESA**'s average is smaller than the average of those algorithms.

3.4 Scenario 4: All Half but One

In this scenario we consider a case with ten Bernoulli arms, considered as being hard: The optimal arm has expectation 0.51 while all the others have expectation 0.5. The results of this experiment are shown in Table 4 and Figure 4. We note that **BESA** gets a smaller average regret than its competitors, and is not beaten by them more than 42% of the time. Thus **BESA** performs best in this hard setting.

3.5 Scenario 5: Truncated Exponential

In order to further demonstrate the flexibility of the **BESA** algorithm, we consider in this scenario the case of rewards coming from an exponential distribution. Five arms

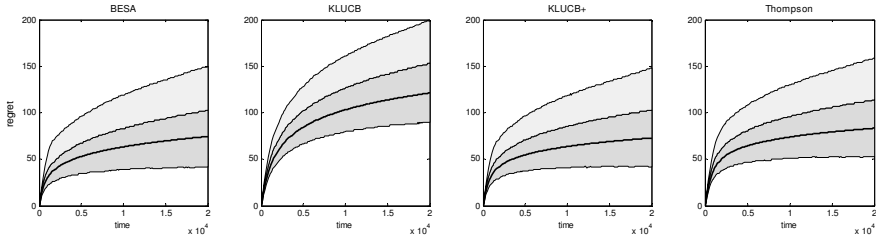

Fig. 3. Regret against time for scenario 3

Table 3. Performance measures for $T = 20,000$ for scenario 3. $\alpha_{1/2}(M,1) = O(0.5025^M)$.

	BESA	**KL-UCB**	**KL-UCB+**	**Thompson sampling**
Average regret at T	74.41	121.21	72.84	83.36
Beat **BESA**	$--$	1.57%	35.41%	3.09%
Average running time	13.85X	2.83X	3.08X	X

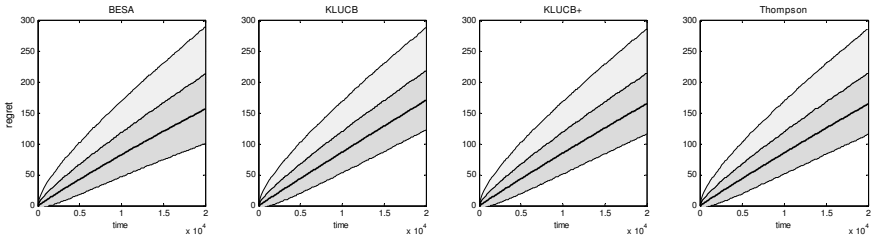

Fig. 4. Regret against time for scenario 4

Table 4. Performance measures for $T = 20,000$ for scenario 4. $\alpha_{1/2}(M,1) = O(0.75^M)$.

	BESA	**KL-UCB**	**KL-UCB+**	**Thompson sampling**
Average regret at T	156.7	170.82	165.28	165.08
Beat **BESA**	$--$	41.36%	41.57%	40.78%
Average running time	19.64X	2.78X	2.96X	X

were considered with parameters $\left\{\frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\right\}$, truncated at 10 then divided by 10 (thus they are bounded in $[0,1]$). The results of this experiment are shown in Table 5 and Figure 5. Note that the regret of **KL-UCB**-exp, which is the version of **KL-UCB** specifically tuned for exponential families and achieving the state-of-the-art for this case, is lower than that of **BESA** only on 5.72% of the runs. Note that **BESA** need not know that the distributions are exponential, that is, we use exactly the same algorithm. Now, as can be seen in the figure of **BESA** the graph is not smooth: the reason is that **BESA** misses the optimal arm if the first reward that it gives is too low. In order to get a smoother behavior, we ran a slightly modified version of **BESA** to skip these corner cases: The modified algorithm is called **BESAT**, and simply pulls each arm ten times before starting running the regular **BESA**. As can be seen in the results this improved the regret dramatically. Now **KL-UCB**-exp beats **BESAT** only on 1.38% of the runs, and similar numbers is achieved for **UCB**-tuned. In [11] a similar scenario is considered, with the difference that they didn't divided the reward by 10. It is actually easy to prove that both **BESA** and **BESAT** are actually invariant by rescaling, that is they pull the same arms in the same order wither we divide the reward by 10 or not. Thus, running the algorithms with the same runs without dividing the rewards by 10, the regret of **BESA** is 532.6 and the one of **BESAT** is 314.1, which is still better than the regrets reported in [11] at Figure 3 (they are above 600).

3.6 Scenario 6: Truncated Poisson

In this scenario we consider the case of Poisson rewards, six Poisson arms are considered with parameters $0.5 + \frac{i}{3}$, where $1 \leqslant i \leqslant 6$, truncated at 10 then divided by 10. A similar scenario was considered in [9] where **KL-UCB**-Poisson is the leading algorithm. From Table 6, **BESA** and **BESAT** outperform **KL-UCB** and **KL-UCB**-Poisson on 95.95% of the runs for **BESA** and 97.99% for **BESAT**, with a much smaller average regret.

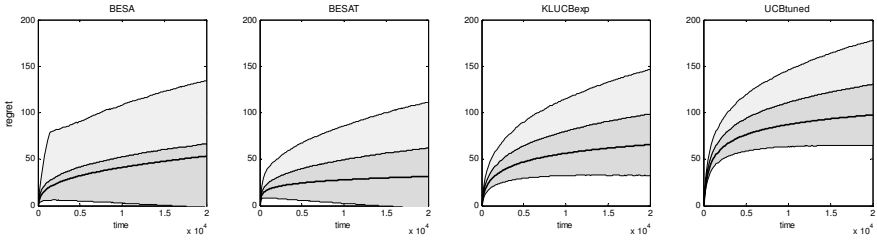

Fig. 5. Regret against time for scenario 5

Table 5. Performance measures for $T = 20,000$ for scenario 5

	BESA	**BESAT**	**KL-UCB**-exp	**UCB**-tuned
Average regret at T	53.26	31.41	65.67	97.6
Beat **BESA**	$--$	40.59%	5.72%	4.33%
Beat **BESAT**	59.41%	$--$	1.38%	0.85%
Average running time	6.01X	7.09X	2.76X	X

3.7 Scenario 7: Uniform Distributions

In this experiment, we consider a challenging setting where arms are uniformly distributed with $X_a \sim \mathcal{U}([0.2, 0.4])$ and $X_\star \sim \mathcal{U}([0, 1])$. Note that there is no natural **KL-UCB** nor **Thompson sampling** algorithm to deal with such family of distribution. In such a scenario, we note that $\alpha_{1/2}(M, 1)$ does not decay exponentially to 0 with M, indicating that this is a difficult scenario for **BESA**. However, it holds that

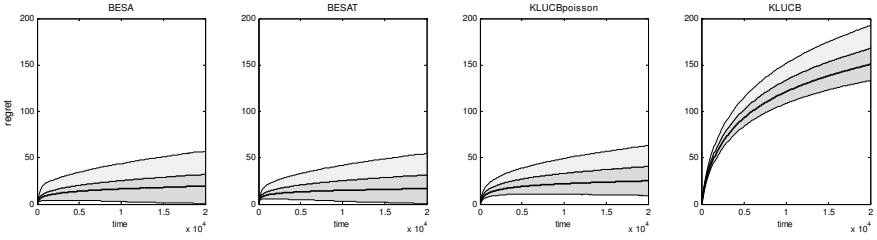

Fig. 6. Regret against time for scenario 6

Table 6. Performance measures for $T = 20,000$ for scenario 6

	BESA	**BESAT**	**KL-UCB**-Poisson	**KL-UCB**
Average regret at T	19.37	16.72	25.05	150.56
Beat **BESA**	$--$	39.92%	4.05%	0.72%
Beat **BESAT**	59.51%	$--$	2.01%	0.17%
Average running time	3.53X	3.49X	1.15X	X

Table 7. Performance measures for $T = 20,000$ for scenario 7

Alg.	**UCB**	**KL-UCB**	**Thompson sampling**
Average regret	21.23	20.72	13.18
Beat **BESA** $n_0 = 0$	24.26%	24.28%	24.7%
Beat **BESA** $n_0 = 3$	7.27%	7.3%	7.83%
Beat **BESA** $n_0 = 7$	1.56%	1.58%	1.76%
Beat **BESA** $n_0 = 10$	0.62%	0.63%	0.74%

BESA	$n_0 = 0$	$n_0 = 3$	$n_0 = 6$	$n_0 = 7$	$n_0 = 8$	$n_0 = 9$	$n_0 = 10$
Average regret	920.12	213.44	50.6	35.38	25.88	17.85	15.42

$\alpha_{1/2}(1, n) = O(\beta^n)$ with $\beta = 0.2$. According to Theorem 1, we should initialize **BESA** by pulling n_0 times each arm before applying **BESA** (that is, run **BESAT**), where for $T = 20,000$, $n_0 \simeq 6.15$. We ran **BESAT** with different number of initialization pulls $n_0 \in \{0, 3, 6, 7, 8, 9, 10\}$, and we also ran **UCB** **KL-UCB** and **Thompson sampling** (with Beta(1,1) prior) on the same set of arms. The average regrets of each of the algorithms in addition to the percentage on which each non-**BESA** algorithm beats **BESA** with different n_0 are provided in Table 7. The average regret of **BESA** improves with increasing n_0 as expected, and as can be seen from the table, **BESA** with $n_0 = 10$ gave a lower average regret than **UCB** and **KL-UCB** and a bit higher than **Thompson sampling** and it was beaten by the other algorithms on less than 0.8% of the runs.

3.8 Summary of the Experimental Results

From the first four numerical experiments, we deduce that **BESA** is able to compete with the state-of-the-art bandit algorithms in the Bernoulli case. It becomes especially good when the gaps are *small*, or when the Bernoulli parameters are small, which are two main cases of practical interest (especially in web-advertising). Scenario 5, 6 and 7 highlight the flexibility of the algorithm: the same algorithm competes favorably against one of the best algorithm for exponential distributions as well as for Poisson distributions. Using a slight modification, we can beat them by an even larger margin.

4 Intuition and Properties

In this section, we provide an explanation for the striking performance of the **BESA** algorithm, and discuss further its advantages and drawbacks. In the next section, we use this intuition to derive a regret bound for the **BESA** algorithm.

4.1 Why Does It Work?

To give intuition why **BESA** works, let us focus on the two-arm bandit problem. The heuristic idea behind the algorithm is that a comparison between two empirical mean estimates built on a very different number of samples n_a and n_b is not really "fair", and that it seems more natural to compare empirical means based on the same number of samples. Thus the algorithm we introduce is based on sub-sampling. In the rich

sub-sampling literature, the works of [7] and [21], show that using sub-sampling without replacement ensures convergence guarantees in a strictly broader setting than sub-sampling with replacement (a.k.a bootstrap). This may provide some informal support for the soundness of the method. However we now provide a more direct justification for the striking performance of **BESA**.

On the theoretical side, one can justify the intuition by looking at the probability of repeatedly choosing a wrong action. If $\mu_b > \mu_a$ and the number of plays of each arm satisfies $n_a > n_b$, the probability that **BESA** chooses a wrong action is approximatively

$$\mathbb{P}\left[\widehat{\mu}\Big(X^a_{1:n_a}\big(I(n_a;n_b)\big)\Big) > \widehat{\mu}\Big(X^b_{1:n_b}\Big)\right], \tag{3}$$

where $I(n_a;n_b) \sim \texttt{Wr}(n_a;n_b)$, and the probability of making M consecutive mistakes is essentially

$$\mathbb{P}\left[\forall m \in [M]\ \widehat{\mu}\Big(X^a_{1:n_a^{(m)}}\big(I_m(n_a^{(m)};n_b)\big)\Big) > \widehat{\mu}\Big(X^b_{1:n_b}\Big)\right], \tag{4}$$

where for all $m \leqslant M$, $I_m(n_a;n_b) \sim \texttt{Wr}(n_a;n_b)$, and where we introduced for convenience the short-hand notation $n_a^{(m)} = n_a + m - 1$.

Now, for deterministic[1] n_a, n_b, (3) typically scales with $\exp\Big(-2n_b(\mu_b-\mu_a)^2\Big)$, by a standard Hoeffding inequality since n_b samples are involved. On the other hand, (4) can decrease at a much faster rate, intuitively of order $\exp\Big(-2n_b\tilde{M}(\mu_b-\mu_a)^2\Big)$ where $\tilde{M}$ is the number of non-overlapping sub-samples of size n_b. Indeed if $I_{m'}(n_a^{(m)};n_b) \cap I_m(n_a^{(m)};n_b) = \emptyset$ for all $m \neq m' \in \mathcal{M} \subset [M]$ where $|\mathcal{M}| = \tilde{M}$, then the corresponding empirical means are independent from each other, which leads to the intuitive improvement. Using sub-sampling, the later event is of high probability for a reasonable $\tilde{M}$ provided that n_a/n_b is large enough. Note that in the case when we do not resort to sub-sampling, such a phenomenon will not happen, due to the strong dependency between the samples at two consecutive time steps. Thus $\tilde{M}$ is essentially 1 which means that the probability of committing M successive mistakes, will stay big, of the order of $\exp\big(-2n_b(\mu_b-\mu_a)^2\big)$. Now in an ideal case, with only $M = n_a/n_b$ subsets, we might get a mistake error scaling with $\exp\big(-2n_a(\mu_b-\mu_a)^2\big)$, even though b has only been pulled n_b times. As long as $n_a/n_b \ll n_a - n_b$, then we need less trials than the procedure based only on confidence interval estimates before accurately discarding the wrong arm with the same probability. This intuitive idea is formalized and captured by Lemma 1, which we consider to be the key for the current analysis.

4.2 Properties of the Algorithm

We now highlight some of the main properties of **BESA**.

[1] The exact argument needs to deal with the fact that $n_a = N_t(a)$ and $n_b = N_t(b)$ are both random stopping times.

Simplicity. We first note the simplicity of the **BESA** compared to previous methods, such as **KL-UCB** for instance that requires some fancy linear program in order to compute the upper confidence bound, or **Thompson sampling** that requires to be able to find an appropriate conjugate prior and implement the update of its parameters, or even the **UCB**-type strategies that generally require some free parameter to be adjusted. Here, **BESA** requires no parameter tuning, and no complicated prior/posterior relation is needed either. Thus the algorithm is directly applicable in a broad range of situation.

Flexibility. A striking property of the algorithm is its flexibility to adapt to various situations. For instance, note that **BESA** does not need to know the support of the distributions, and is also invariant under rescaling. This is not the case of most bandit algorithms that explicit use the knowledge of the support $[a, b]$ of distributions. We believe this can be a serious advantage in some situation. Moreover, both **Thompson sampling** and **KL-UCB** are dependent on a considered parametric set of distributions: in practice a different set of distribution leads to a different implementation. **BESA** does not need such parameters, keeps the same form in all situations, and more importantly still achieves excellent performance in a number of situations, as detailed in Section 3.

Efficiency. Finally, one might wonder about the computationally efficiency of **BESA** due to the use of a sub-sampling method, that is generally not memory less. We were a bit worried about this fact, and thus we implemented the algorithm in a naive way and reported the computational cost of the algorithm in each table for completeness. We conclude from these results that the computational cost of the algorithm is essentially not a problem. Moreover, note that, due to the i.i.d. nature of the data, one may use fancier but more efficient sub-sampler techniques. A naive implementation needs to save all the received rewards. In case the rewards take only finitely many values one can use instead a counter for each possible value. To avoid distracting the reader from the main message, we do not discuss possible tricks that could be used to improve further the numerical efficiency of the method.

5 Regret Upper-Bound

In this section, we provide a simple regret analysis of the **BESA** algorithm[2]. Formalizing further the heuristic intuition of Section 4, it is actually possible to derive a non-trivial regret bound for the **BESA** algorithm, given in Theorem 1. In order to characterize the difficulty of a bandit problem, we now define the following problem-dependent quantity

Definition 1. *For integers M, n and $\lambda \in [0, 1]$, we define the balance function of the distributions $(\nu_a, \nu_\star)$ as*

$$\alpha_\lambda(M, n) = \mathbb{E}_{Z\sim\nu_{\star,n}}\left[\left(1 - F_{\nu_{a,n}}(Z) + \lambda\nu_{a,n}(Z)\right)^M\right],$$

where $\nu_{a,n}$ is the distribution of $\sum_{i=1}^n X_i^a$ with $X_i^a \overset{i.i.d}{\sim} \nu_a$ and F_ν is the cdf of ν (that is, $F_\nu(x) = \mathbb{P}_{X\sim\nu}(X \leqslant x)$).

[2] We study a slightly modified version, that break ties uniformly at random.

Let us provide some intuition on two examples. Note that α_λ is not increasing both in M and n. First, let us consider two Bernoulli arms with $X^a \sim \mathcal{B}(\mu_a)$ and $X^\star \sim \mathcal{B}(\mu_\star)$. We can compute easily that $\alpha_\lambda(M,1) = (\lambda\mu_a)^M\mu_\star + \big(\lambda + \mu_a(1-\lambda)\big)^M(1-\mu_\star)$. Now, since $\mu_a < \mu_\star \leqslant 1$, we deduce that $\alpha_{1/2}(M,1) \overset{M\to\infty}{\to} 0$, and more precisely that

$$\alpha_{1/2}(M,1) = O\bigg(\Big(\frac{\mu_a \vee (1-\mu_a)}{2}\Big)^M\bigg).$$

Thus it converges exponentially fast to 0. Note, however that $\alpha_1(M,1) \overset{M\to\infty}{\to} 1-\mu_\star$, which is non zero unless $\mu_\star = 1$. Second, let us consider the case of two Uniform arms $X^a \sim \mathcal{U}([0.2, 0.4])$ and $X^\star \sim \mathcal{U}([0,1.])$. For all λ it holds that $\alpha_\lambda(M,n) \overset{M\to\infty}{\to} 0.2^n$. Thus there is no exponential decay with M. However, it holds that $\alpha_\lambda(1,n) = O(0.2^n)$.

We now prove the following

Theorem 1. *Let $\mathcal{A} = \{\star, a\}$ be a two-armed bandit with bounded rewards in $[0,1]$, and $\Delta = \mu^\star - \mu_a$ be the mean gap. Let us moreover assume that there exists $\alpha \in (0,1)$ and $c > 0$ such that $\alpha_{1/2}(M,1) \leqslant c\alpha^M$. Then the regret of **BESA** at time T is controlled by*

$$\mathfrak{R}_T \leqslant \frac{11\log(T)}{\Delta} + C_{\nu_a,\nu_\star} + O(1)\,,$$

where $C_{\nu_a,\nu_\star}$ depends on the parameters of the problem α, c and Δ, but not on T. Moreover if there exists some $\beta \in (0,1)$ and $c > 0$ such that $\alpha_{1/2}(1,n) \leqslant c\beta^n$. Let us define

$$n_{0,T} = \left\lceil \frac{\ln(T) - \ln\big((1-\beta)C\big)}{\ln(1/\beta)} \right\rceil .$$

*Then if **BESA** is initialized with $n_{0,T}$ pulls of each arm, then its regret at time T is controlled by*

$$\mathfrak{R}_T \leqslant \frac{11\log(T)}{\Delta} + n_{0,T} + \tilde{C}_{\nu_a,\nu_\star} + O(1)\,.$$

where $\tilde{C}_{\nu_a,\nu_\star}$ depends on C and on the parameters β, c and Δ, but not on T.

Remark 1. Up to Lemma 1 (see below) which is a purely probabilistic result, and is independent on the bandit setting, the proof is arguably simpler than the typical ones used for Thompson sampling. In particular, we do not need to resort to a fancy "Bernoulli-Beta" trick that is used in classical proofs of Thompson sampling and does not extend easily to general distributions (see for instance [17] that is entirely devoted to the extension to exponential families of dimension one).

Remark 2. Since one needs not use empirical confidence intervals in the analysis, but simply confidence intervals, one can hope to derive much tighter results in the future, using Kullback-Leibler-based Chernoff bounds, or event the sharpest Sanov bounds.

We provide the full proof of this result in the appendix. It mainly follows the proof of [16] that provides a sharp analysis of the **Thompson sampling** algorithm, with some simplifications: first, we consider only two arms, which enables us to skip a recurrence argument (but the same technique could be used to extend our analysis to the case of K-arms); then, we only use mean-based arguments essentially for clarity of exposure. We believe it is more important at this point to provide a clear intuition about why the algorithm works than to provide a tight analysis based on Kullback-Leibler concentration results that are trickier to catch. Now for clarity, we summarize in the next Lemma what we believe is the key result for the regret analysis of **BESA**. This purely probabilistic result is specific to the properties of the sub-sampling procedure.

The sketch of proof is as follows: As usual, we express the regret in terms of the expected number of pulls of sub-optimal arms, that are further decomposed according to the event that the optimal arm has been pulled enough or not. Under the event that the optimal arm is pulled enough, we control easily the probability of mistake resorting to standard proof techniques based on concentration bounds. One difference with respect to standard bounds is that we use here a Serfling-Hoeffding ([22]) concentration inequality. This gives the first term of the regret. The next and difficult step is to show, as usual, that the optimal arm is indeed pulled enough with high probability. To that end, we borrow a proof technique considered in [16]: we introduce the random times τ_j between the j^{th} and $(j+1)^{th}$ pull of the optimal arm, and show that they cannot be too large for too many j, that is to show that the number of *consecutive* mistakes made by the algorithm must be small with high probability. This is one key of the regret analysis of **Thompson sampling** that enables to derive an optimal performance bound. The novelty that we introduce for the analysis of **BESA** is to relate this number of consecutive mistakes to the probability that many sub-samples of small size do not overlap, as explained in Section 4. The precise lemma that covers this part, and that eventually leads to our regret bound is the following:

Lemma 1 (Maximal non-overlapping sub-samples). *Let $\mathcal{M} = \{p, \dots, q\} \subset \mathbb{N}$ be some interval. Let $j, M \in \mathbb{N}$ be such that $p \geqslant 2j$, and $M \leqslant |\mathcal{M}|$. For all $s \in \mathcal{M}$, we introduce the random variable $I_s(s-j;j) \sim \mathtt{WR}(s-j;j)$. Then the function defined by*

$$f_{\mathcal{M}}(M,j) = 1 - \mathbb{P}\Big[\exists s_1 < \dots < s_M \in \mathcal{M},$$

$$\forall m \neq m' \in [M] : I_{s_m}(s_m - j; j) \cap I_{s_{m'}}(s_{m'} - j; j) = \emptyset\Big]$$

is decreasing with p. Moreover, for a sequence of intervals $\mathcal{M}_t = \{p_t, \dots, q_t\}$, such that $\lim_{t\to\infty} \frac{q_t - p_t}{t} = C > 0$ and integers $M_{t,j}$ such that $M_{t,j} j = O(\ln(q_t))$, we have

$$f_{\mathcal{M}_t}(M_{t,j}, j) = o(t^{-1})\,. \tag{5}$$

One way to show this lemma is by studying $f_{\mathcal{M}}(M,j)$ formally, and trying to see how it behaves with the different parameters. This however turns out to be tedious and

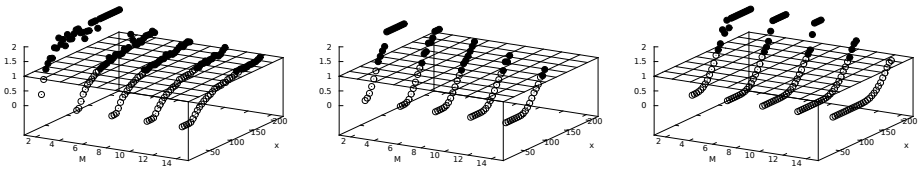

Fig. 7. α as a function of x and M for $j = 3$ (left), $j = 8$ (middle) and $j = 15$ (right). (Note that for $\Delta = 0.3$ and $T = 20,000$, then $\frac{1}{2\Delta}\log(T) \simeq 16.5$.) The black circles indicate when $\alpha_x > 1$ and the white circles when $\alpha_x < 1$.

very technical. On the other hand, we can take advantage of the fact that this function is problem-independent and thus can be computed off-line. It can actually be simulated, and since it is decreasing with $|\mathcal{M}|$, it is enough to study its behavior for small $|\mathcal{M}|$. For our purpose in the analysis, we only need to look at small values of $M = O(\log(T))$ (note that for a time horizon $T = 20,000$, then $\log(T) < 10$). Similarly we use small value for $n_0 \leqslant j \leqslant u_T = O(\log(T))$ as well. It is not difficult to simulate $f_{\mathcal{M}}(M, j)$ for $\mathcal{M}_x = [2j + 1, x]$ with various values of M and x. We are interested in the ratio $\alpha = -\log(f_{\mathcal{M}_x}(M, j))/\log(x)$, and observe numerically that this ratio is increasing with x and quickly becomes larger than 1, that is $f_{\mathcal{M}_x}(M, j) < x^{-1}$ for large enough $x \geqslant C_M = o(M)$ that is slowly increasing with M. In the analysis of Theorem 1, we use $x = O(t/\log(t))$, and $M = O(\log(T))$, and thus as soon as $t \geqslant C'\log(t)$, which happens for $t \geqslant C$ for some numerical constant C, then $f_{\mathcal{M}_t}(M, j)$ starts decaying faster than t^{-1}, that is $f_{\mathcal{M}_t}(\log(t)/j, j) = o(t^{-1})$. In figure 7, we plot the function α in terms of x and M, and for different values of j. Each point is the result obtained via $5,000$ replications. We report especially in blue the regions when α becomes larger than 1, which is the critical value to ensure that the lemma holds.

6 Discussion and Conclusion

In this paper, we introduced a novel algorithm for the stochastic multi-armed bandit that is based on *sub-sampling*. We provided a careful experimental analysis of the **BESA** algorithm, by comparing it with the optimized versions of the state-of-the-art algorithms known in each situation. We demonstrated the advantage of **BESA** specifically in the case of Bernoulli distributions, including the case of small parameters and small gaps, as well as exponential and Poisson distributions. For completeness, we reported three measures of performance of the algorithm: plots of the cumulative reward, included quantiles, the percentage it is beaten by other standard algorithms and the numerical complexity with respect to the fastest method.

The algorithm has several striking properties: it is simple to implement, and is very flexible. It does not need to know a set of distributions in advance, unlike **Thompson sampling** or **KL-UCB** and does not even need to know the support, unlike **UCB** or **kl-UCB**. It is also invariant under rescaling of the rewards. This is thus a fully non-parametric algorithm, that competes favorably against standard algorithms.

We finally provide a regret analysis for **BESA**, which shows that the regret of the algorithm is logarithmic (we believe that the constants are not tight). More importantly, we provided a novel proof technique that we believe conveys the core intuition why the algorithm is working, and can lead to much tighter bounds in the future.

Now that we have introduced this algorithm and shown its flexibility, it seems natural to try to extend the **BESA** to other settings. One first direction of research is to consider the *contextual*-bandit problem, another one is to consider the *adversarial* multi-armed bandit setting.

Acknowledgements. This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 306638 (SUPREL) and the Technion.

References

[1] Agrawal, R.: Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. Advances in Applied Probability 27(4), 1054–1078 (1995)
[2] Agrawal, S., Goyal, N.: Further optimal regret bounds for thompson sampling. In: International Conference on Artificial Intelligence and Statistics, Scottsdale, AZ, US. JMLR W&CP, vol. 31 (2013)
[3] Audibert, J.-Y., Bubeck, S.: Minimax policies for adversarial and stochastic bandits
[4] Audibert, J.-Y., Munos, R., Szepesvári, C.: Exploration-exploitation trade-off using variance estimates in multi-armed bandits. Theoretical Computer Science 410, 1876–1902 (2009)
[5] Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research 3, 397–422 (2003)
[6] Auer, P., Ortner, R.: UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. Periodica Mathematica Hungarica 61(1-2), 55–65 (2010)
[7] Bickel, P.J., Sakov, A.: On the choice of m in the m out of n bootstrap and confidence bounds for extrema. Statistica Sinica 18, 967–985 (2008)
[8] Burnetas, A.N., Katehakis, M.N.: Optimal adaptive policies for sequential allocation problems. Adv. Appl. Math. 17(2), 122–142 (1996)
[9] Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., Stoltz, G.: Kullback–leibler upper confidence bounds for optimal sequential allocation. Ann. Statist. 41(3), 1516–1541 (2013)
[10] Chang, F., Lai, T.L.: Optimal stopping and dynamic allocation. Advances in Applied Probability 19(4), 829–853 (1987)
[11] Garivier, A., Cappé, O.: The KL-UCB algorithm for bounded stochastic bandits and beyond. In: Proceedings of the 24th annual Conference on Learning Theory, COLT 2011 (2011)
[12] Gittins, J.C.: Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society. Series B (Methodological) 41(2), 148–177 (1979)
[13] Gittins, J.C., Jones, D.M.: A dynamic allocation index for the discounted multiarmed bandit problem. Biometrika 66(3), 561–565 (1979)
[14] Gittins, J.C., Weber, R., Glazebrook, K.: Multi-armed Bandit Allocation Indices. Wiley (1989)
[15] Honda, J., Takemura, A.: An asymptotically optimal bandit algorithm for bounded support models, pp. 67–79
[16] Kaufmann, E., Korda, N., Munos, R.: Thompson sampling: an asymptotically optimal finite-time analysis. In: Bshouty, N.H., Stoltz, G., Vayatis, N., Zeugmann, T. (eds.) ALT 2012. LNCS (LNAI), vol. 7568, pp. 199–213. Springer, Heidelberg (2012)

[17] Korda, N., Kaufmann, E., Munos, R.: Thompson sampling for 1-dimensional exponential family bandits. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) NIPS, Lake Tahoe, Nevada, United States, vol. 26, pp. 1448–1456 (2013)
[18] Lai, T.L., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics 6(1), 4–22 (1985)
[19] Maillard, O.-A., Munos, R., Stoltz, G.: Finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. In: Proceedings of the 24th Annual Conference on Learning Theory, COLT 2011 (2011)
[20] Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the American Mathematics Society 58, 527–535 (1952)
[21] Romano, J.P., Shaikh, A.M.: On the uniform asymptotic validity of subsampling and the bootstrap. The Annals of Statistics 40(6), 2798–2822 (2012)
[22] Serfling, R.J.: Probability inequalities for the sum in sampling without replacement. The Annals of Statistics 2(1), 39–48 (1974)
[23] Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25, 285–294 (1933)
[24] Thompson, W.R.: On the theory of apportionment. American Journal of Mathematics 57, 450–456 (1935)