JMLR: Workshop and Conference Proceedings 25:129–143, 2012 Asian Conference on Machine Learning

A stochastic bandit algorithm for scratch games

Raphaël Féraud RAPHAEL.FERAUD@ORANGE.COM and
Tanguy Urvoy TANGUY.URVOY@ORANGE.COM
Orange Labs, 2, avenue Pierre Marzin, 22307 Lannion, France

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

Stochastic multi-armed bandit algorithms are used to solve the exploration and exploitation dilemma in sequential optimization problems. The algorithms based on upper confidence bounds offer strong theoretical guarantees, they are easy to implement and efficient in practice. We considers a new bandit setting, called "scratch-games", where arm budgets are limited and reward are drawn without replacement. Using Serfling inequality, we propose an upper confidence bound algorithm adapted to this setting. We show that the bound of expectation to play a suboptimal arm is lower than the one of UCB1 policy. We illustrate this result on both synthetic problems and realistic problems (ad-serving and emailing campaigns optimization).

Keywords: Multi-armed bandits, Serfling inequality, drawing without replacement

1. Introduction

In its most basic formulation, the stochastic multi-armed bandit problem can be stated as follows: there are K arms, each having a fixed, unknown, and independent probability-law of reward. At each step, a player chooses an arm and receives a reward. This reward is drawn according to the selected arm's law and it is independent of previous actions. Under this assumption, many policies have been proposed to optimize the long-term cumulated reward. We study here the case where the reward of each arm is drawn without replacement from a finite pool. In this setting, which we call scratch-game, the number of actions of an arm is limited and the assumption that actions are independent is wrong. Is it possible to adapt standard stochastic bandit algorithms to take advantage of this non-replacement setting?

The answer of this question is interesting by itself: it can open new doors from a theoretical point of view. But the real motivation is operational: the non-replacement assumption is realistic for several real-world problems like online advertising or optimization of marketing campaigns. In the pay-per-click model for online advertising, the contract between advertiser and publisher includes condition to provide gradual revenues depending on click-through rate (CTR). In this case when placing ads on web pages, the interest of the publisher is to maximize the CTR. Each ad in each context (position on web pages, user profiles...) is an arm. The reward is the click on the banner. Usually, the number of displays of each ad is limited according to a contract called "capping". It is also desirable, in order to respect the user's weariness in face of advertising solicitation and to dispatch user's profiles on different ads, to limit the number of repetitions of the same banner. Theses two kinds of limitations

are well modeled by the non-replacement assumption. For a telecommunication operator, marketing campaigns are regularly deployed to promote new services to customers. For each customer, a profile is built from his past behavior, his supposed or evaluated appetency for each service, or a marketing segmentation. For each service, several hypothetic campaigns are considered. Each marketing campaign has a finite budget and a media to contact customers. Each marketing campaign on each profile is an arm. The reward is a subscription. As the budgets are limited, and as a customer cannot subscribe twice to the same service, the draw of an arm is without replacement.

In this paper, we formalize and study multi-armed bandits for drawing without replacement. We provide a new upper bound using Serfling inequality Serfling (1974). We show that in the case of drawing without replacement the bound of the expectation of playing a suboptimal arm is lower than the one of UCB1 policy Auer et al. (2002). The efficiency of the proposed algorithm is validated on synthetic problems, on an ad serving application and on optimization of emailing campaigns.

2. Related works

This section provides a short overview on stochastic multi-armed bandit problem related to our work. In the stochastic bandit problem, each arm delivers rewards that are independently drawn from an unknown distribution. Efficient solutions based on optimism in the face of uncertainty principle have been proposed Agrawal (1995); Lai and Robbins (1985). They compute an index for each arm and they choose the arm with the highest index. It can be showen that the regret, the cumulative difference between the optimal reward and the expectation of reward, is bounded by a logarithmic function, which is the best possible. Subsequent work introduced simpler policies, based on an upper confidence bound to compute the index, that achieve logarithmic bound uniformly Auer et al. (2002). Recently, different versions of these kinds of policies have been proposed, to take into account the observed variance Audibert et al. (2009), the tree structure of arms Kocsis and Szeoesvàri (2006); Bubeck et al. (2008), based on the Kullback-Leibler divergence to compute the index Garivier and Cappé (2011), or based on a Bayesian approach to compute the index Kaufman et al. (2012). Our work is an adaptation of these classes of policies for drawing without replacement.

Recently some authors have proposed multi-armed bandit with dependent arms Pandey et al. (2007b,a); Varsha Dani (2008). The purpose is to increase the accuracy by exploiting the similarity between arms. In our case the draws of an arm are dependent and we assume that arms are independent. To take into account finite inventory for online-advertising, it is possible to model the lifetime of arms Chakrabarti et al. (2008). When a new campaign is displayed a new arm is born. When the budget of a campaign reaches zero, the corresponding arm is dead. Chakrabarti et al. (2008) propose efficient policies when the distribution of reward and expected lifetime are known. An extension of UCB1 algorithm, called UCB1 K/C, is also proposed to take into account the birth and death of arms. This policy does not need to know the distribution of reward. In their experiments, the authors exhibit better results for UCB1 K/C than for UCB1. However the proposed policies to model the lifetime

of arms do not take into account the dependency between draws. Our work focuses on this dependency. Moreover, the proposed upper bound can be used with UCB1 K/C.

3. Problem setup: scratch games

In the multi-armed bandit setting, to maximize his gain the player has to find the best game as soon as possible, and then to exploit it. In the scratch game setting, the number of tickets is bounded. When the player has found the best game, he knows that this game will die. The player needs to re-explore before the best game die to find the next best game. The usual tradeoff between exploration and exploitation has to be revisited. In this section, we define and we formalize the scratch game setting.

We consider a set of K scratch games. Each game i has a finite number of tickets N_i, and $N = \sum_i N_i$. For each game i, the rewards of each ticket have been drawn independently and identically according to an unknown probability law. Let s_{ij} be the random variable giving the reward of the j-th tickets for the game i. For any $j > N_j$, all the tickets have been scratched: we have $s_{ij} = 0$.

At each time step t, the player chooses a scratch game i and received the reward $s_{i,n_i(t)}$, where $n_i(t)$ is the number of tickets scratched of game i at time t. In the following to simplify the notations, we will use $s_i(t)$ for $s_{i,n_i(t)}$. To make his choice, the player knows for each game the initial number of tickets N_i, and the sequence of past rewards $S^{t-1} = s_{i_1}(1), s_{i_2}(2), ..., s_{i_{t-1}}(t-1)$, where i_t is the scratch game chosen at time t.

At time t, a policy P allows to choose the next scratch game i_t. A policy can be stochastic, if the next game is chosen according to a probability, static if all games are chosen at the beginning, or dynamic if the choice of the next game is determined by past rewards. In the following, we will consider the case of deterministic policies: static or dynamic. A dynamic policy can be defined as a function such that:

$$P : S^{t-1} \mapsto [K]$$

By applying a policy P, at time t a sequence of choices is obtained $(i_1, i_2, ..., i_t) \in [K]^t$. At time t, the gain of the policy P is:

$$G_P(t) = \sum_{k \leq t} s_{i_k}(k)$$

The policy P is deterministic. However, the tickets or rewards $s_i(t)$ have been drawn according to unknown distributions. The expected gain of the policy P on all the sequences of scratched tickets $s_{i_1}(1), s_{i_2}(2), ..., s_{i_t}(t)$ is:

$$E[G_P(t)] = \sum_{i=1}^{K} E[G_i(t)] = \sum_{i=1}^{K} E[n_i(t)].\mu_i$$

Where $G_i(t)$ is the cumulated reward of game i, and μ_i is the mean reward of the game i. Knowing for each scratch game the number of tickets N_i , the number of tickets scratched $n_i(t)$, and the cumulated reward $G_i(t)$, we would like to find an efficient policy P in order to optimize uniformly the expected gain $E[G_P(t)]$ on all the sequences of draws for all $t < N$. Notice that the number of tickets is finite and thus if all the tickets are scratched, all the

policies are equivalent. In the following, we focuse on the case where $t \ll N$. For illustration purpose, suppose that we have two scratch games with binary reward:

- game one has 20 tickets, with 10 winning tickets,
- game two has 100 tickets, with 11 winning tickets.

Knowing the initial mean reward of both games, the optimal policy is to play the game one while its remaining mean reward is higher than 11/100 and then to play the game with the highest remaining mean reward. Let $Q_P(t)$ be the set of remaining games at time t using the strategy P. We can define the optimal policy O as the policy which plays at step t the game with the highest remaining mean reward:

$$\arg \max_{i \in Q_O(t)} \frac{N_i.\mu_i - n_i(t).\hat{\mu}_i(t)}{N_i - n_i(t)} \tag{1}$$

For practical use, the optimal policy raises a problem. Indeed, the optimal policy exploits the difference between the mean reward μ_i of each game i and the estimated reward $\hat{\mu_i}(t)$ to choose the best remaining game. However the mean reward is unknown. The only way to evaluate it is to use $\hat{\mu}_i(t)$. Then if we replace μ_i by $\hat{\mu}_i(t)$ in (1), it is straightforward to show that the estimated optimal policy consists in playing the game with the highest estimated mean.

We can consider a good suboptimal policy: play the game one, until all its tickets have been scratched, and then play the game two. It can be defined as following: sort games i by decreasing mean rewards, and then play each game until its number of remaining tickets is zero. This policy is the optimal static policy. We note it OS. For practical use μ_i is unknown. It can be estimated by $\hat{\mu}_i(t)$. As for the optimal policy, the estimated optimal static policy consists in playing the game with the highest estimated mean. In our illustrative example, before the step 21, this policy plays the game one, and after the step 21 it plays game two. We note i_t^* the game chosen by the optimal static policy OS at time t. We define the weak regret R as:

$$R(t) = \sum_{i=1}^{K} E[n_i(t)].(\mu_{i_t^*} - \mu_i)$$

We define $\Delta_i(t)$ as the difference between the mean reward of the game i chosen by a policy and the mean reward of the game chosen by the optimal static policy OS:

$$\Delta_i(t) = \max_{j \in Q_{OS}(t)} \mu_j - \mu_i = \mu_{i_t^*} - \mu_i$$

Notice that the game chosen by the policy OS does not depend on the sequence of scratched tickets. $\Delta_i(t)$ is deterministic. This property will be useful below to provide bounds on $E[n_i(t)]$ in order to compare algorithms.

4. A bandit algorithm for scratch games

We have seen that to build a good policy, we need to evaluate the mean rewards of games in order to sort them. We propose to use the optimism in the face of uncertainty principle to sort the games. UCB Auer et al. (2002) evaluates an upper confidence bound on the mean reward for each game to build an index, and it chooses the game with the highest index. A first strategy to adapt UCB policies to finite inventory is to suppress a scratch game (or arm) when its number of remaining tickets reaches zero. This change prevents for choosing a scratch game with no tickets. In the following, we use this modification for all policies. UCB policies are based on Chernoff-Hoeffding inequality. Its use supposes that the rewards are drawn independently. This assumption does not hold for drawing without replacement. In this case, the Serfling inequality Serfling (1974) can be used. Let $s_1, ..., s_n$ be a sample drawn without replacement from a finite list of values between 0 and 1 $S_1, ..., S_N$, then for all $\epsilon > 0$:

$$P\left(\frac{1}{n}\sum_{t=1}^{n} s_t + \epsilon \leq \overline{S}\right) \leq e^{-\frac{2n\epsilon^2}{1-\frac{n-1}{N}}}$$

We propose to use the Serfling inequality rather than Chernoff-Hoeffding inequality to build an upper confidence bound for scratch game. Let $B_i(t)$ be the index of game i at time t:

$$B_i(t) = \frac{1}{n_i(t)}\sum_{k=1}^{n_i(t)} s_i(k) + \sqrt{\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{2\log t}{n_i(t)}}$$

As for UCB1, The index $B_i(t)$ is used to sort each game at time t. The optimism in face of uncertaintly principle consists in playing the game with the highest index (see Figure 1). We call this policy UCBWR for Upper Confidence Bounds for drawing Without Replacement.

With UCBWR policy, the mean reward is balanced with a confidence interval weighted by the sampling rate of the game. Then, when the number of plays $n_i(t)$ of the scratch game i increases, the confidence interval term decreases faster than with UCB1 policy. The exploration term tends to zero when the sampling rate tends to 1. The decrease of the exploration term is justified by the fact that the potential reward decreases as the sampling rate increases. Notice that if all games have the same sampling rates, the rankings provided by UCB1 and UCBWR are the same. The difference between rankings provided by UCBWR and UCB1 increases as the dispersion of sampling rates increases. We can expect different performances, when the initial numbers of tickets N_i are different. In this case, scratch a ticket from a game with a low number of tickets has more impact on its upper bound than a game with a high number of tickets.

Using Serfling inequality, we see that the probability of mistake decreases quickly to zero:

$$P\left(\frac{1}{n_i(t)}\sum_{k=1}^{n_i(t)} s_i(k) + \sqrt{\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{2\log t}{n_i(t)}} \leq \mu_i\right) \leq e^{-4\log t} = t^{-4} \tag{2}$$

Initialisation: play each gome one, $t = 1$

While $(t \leq n)$

- For all games i in $Q_{UCBWR}(t)$
  - if $n_i(t) = N_i$ then suppress the game i in $Q_{UCBWR}(t)$
  - Evaluate $B_i(t)$
- Play game i in $Q_{UCBWR}(t)$ that maximizes $B_i(t)$
- Receive reward $s_i(t)$
- $n_i(t+1) = n_i(t) + 1$
- $t = t + 1$

Figure 1: UCBWR policy is an adaptation of UCB1 policy for drawing without replacement

Theorem. For all $K > 1$, if policy UCBWR is run on K scratch games corresponding each to a finite list of rewards, then for any suboptimal scratch game i with $N_i > 0$, we have:

$$E[n_i(t)] \leq 8 \left(1 - \frac{E[n_i(t)] - 1}{N_i}\right) \frac{\log t}{\Delta_i^2(t)} + 2$$

The bound obtained by UCBWR policy is lower or equal than the one obtained applying UCB1 policy to scratch game:

- equal at the initialization when the expected sampling rate is zero,
- and lower when the expected sampling rate increases.

Since UCBR is derived from the UCB policies using Serfling inequality in place of Chernoff-Hoeffding inequality, the proof of this theorem is close to the one provided for UCB1 by Auer et al. (2002). We give below the major steps of the proof.

Proof.

Suppose that at time t, the estimated means of reward are in their confidence interval. Then, we have:

$$\mu_i - \sqrt{\left(1 - \frac{n_i(t) - 1}{N_i}\right) \frac{2 \log t}{n_i(t)}} \leq \hat{\mu}_i(t) \leq \mu_i + \sqrt{\left(1 - \frac{n_i(t) - 1}{N_i}\right) \frac{2 \log t}{n_i(t)}} \tag{3}$$

Suppose that at time t, a suboptimal game is chosen. Then we have:

$$B_i(t) \geq B_{i_t^*}(t)$$

$$\Rightarrow \hat{\mu}_i(t) + \sqrt{\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{2\log t}{n_i(t)}} \geq \hat{\mu}_{i_t^*}(t) + \sqrt{\left(1 - \frac{n_{i_t^*}(t)-1}{N_{i_t^*}}\right)\frac{2\log t}{n_{i_t^*}(t)}} \quad (4)$$

Using the inequalities (3) and (4), we have:

$$\mu_{i_t^*} \leq \mu_i + 2\sqrt{\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{2\log t}{n_i(t)}}$$

If the estimated means are in their confidence interval and if a suboptimal scratch game is chosen, we have:

$$n_i(t) \leq \frac{8\left(1 - \frac{n_i(t)-1}{N_i}\right)\log t}{\left(\mu_{i_t^*} - \mu_i\right)^2} = 8\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{\log t}{\Delta_i^2(t)} \quad (5)$$

For all integer u, we have:

$$n_i(t) \leq u + \sum_{k=u+1}^{t} 1\left\{i_k = i; n_i(k) > u\right\}$$

As we supposed that a suboptimal game has been chosen, we have for all integer u:

$$n_i(t) \leq u + \sum_{k=u+1}^{t} 1\left\{\exists n_i(k) : u < n_i(k) \leq k, \exists n_{i_t^*}(k) : 1 < n_{i_t^*}(k) \leq k, B_i(k) \geq B_{i_t^*}(k)\right\} \quad (6)$$

If we choose:

$$u = 8\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{\log t}{\Delta_i^2(t)} + 1$$

Then the inequality (5) does not hold and then if a suboptimal game is chosen at least one of the two inequalities (3) does not hold. Using the inequality (2), this probability is bounded by k^{-4}. If we bound the probability to draw a suboptimal game by the sum of probabilities, we have:

$$n_i(t) \leq 8\left(1 - \frac{n_i(t)-1}{N_i}\right)\frac{\log t}{\Delta_i^2(t)} + 1 + \sum_{k=u+1}^{N}\left[\sum_{s=u+1}^{k} k^{-4} + \sum_{s=1}^{k} k^{-4}\right]$$

If we take the expectation on all sequences of draws $s_{i_1}(1), s_{i_2}(2), ..., s_{i_t}(t)$ of both sides of the inequality (6), as $\Delta_i(t)$ is deterministic we obtain:

$$E[n_i(t)] \leq 8\left(1 - \frac{E[n_i(t)]-1}{N_i}\right)\frac{\log t}{\Delta_i^2(t)} + 1 + \frac{\pi^2}{3}$$

To obtain bounds of UCB1 for the scratch games, we just have to replace Serfling inequalities by Chernoff-Hoeffding inequalities in (3). It corresponds to a sampling rate equal to zero. By replacing it by zero in inequalities (4) and (5) and then in the theorem, the obtained bound is the same than for the stochastic multi-armed bandit problem, with $\Delta_i(t) = \max_{j \in Q_{OS}(t)} \mu_j - \mu_i$.

Corollary. For all $K > 1$, if policy UCBWR is run on K scratch games corresponding each to a finite list of rewards, then for any suboptimal scratch game i with $N_i > 0$, we have:

$$E[n_i(t)] \leq \frac{8 \log t}{\Delta_i^2(t) + \frac{8 \log t}{N_i}} + 1 + \frac{\pi^2}{3}$$

As in the case of UCB policy, for each game the bound is small when the rewards are smaller than the ones provided by the optimal static policy OS, but more, it is small when the number of tickets is small in comparison to the logarithm of the total number of tickets scratched.

Proof. By factoring the expectation from the theorem, we obtain:

$$E[n_i(t)] \leq 1 + \frac{N_i \left(8 \log t + \frac{\pi^2}{3} \Delta_i^2(t)\right)}{N_i \Delta_i^2(t) + 8 \log t}$$

Then:

$$E[n_i(t)] \leq 1 + \frac{\pi^2}{3} + \frac{8 \log t}{\Delta_i^2(t) + \frac{8 \log t}{N_i}}$$

5. Experimental results

5.1. Experimental setup

We have shown that UCBWR policy exhibits lower upper bound to draw a suboptimal game than UCB1 policy. In this section, we compare the behavior of the UCBWR and UCB1 algorithms on some synthetic problems to complete formal analysis. For all policies when the number of tickets of a scratch game reaches zero, it is suppressed from the list of games. In order to compare algorithms, we plot the estimated weak regret versus the number of draws t for UCB1 and UCBWR policies:

$$\hat{R}(t) = \mu_{i_t^*}.t - \sum_{i=1}^{K} \overline{G}_i(t)$$

The gain of the tested policy is evaluated on ten trials. $\mu_{i_t^*}$ is the mean reward of the scratch game chosen by the optimal static policy OS. In order to sum up, we evaluate the mean weak regret (see Table 1):

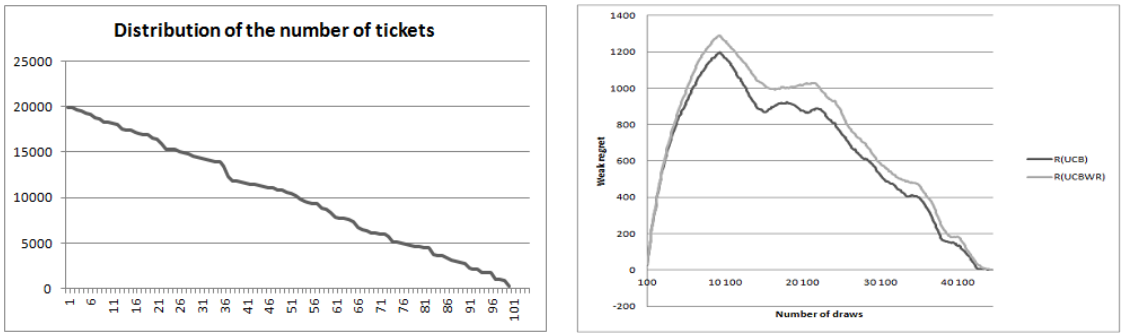
$$\overline{R} = \sum_{t=1}^{N} \hat{R}(t)$$

synthetic problem	Number of tickets	Number of winning tickets	$\overline{R}(UCB1)$	$\overline{R}(UCBWR)$
Uniform distribution	46711	22324	652	730
Quasi-pareto distribution	273817	116374	1485	1177
Pareto distribution	499524	47774	807	677

Table 1: Summary of results on the synthetic problems

5.2. Synthetic problems

In the first synthetic problem 100 scratch games with binary reward are used. Each scratch game has 1 to 1000 tickets drawn from a uniform distribution. For each scratch game, winning tickets are drawn from a Bernoulli distribution, with p parameter drawn from a uniform distribution between 0 and 1. 46711 tickets including 22324 winning tickets spread over 100 scratch games are obtained (see Figure 2a).

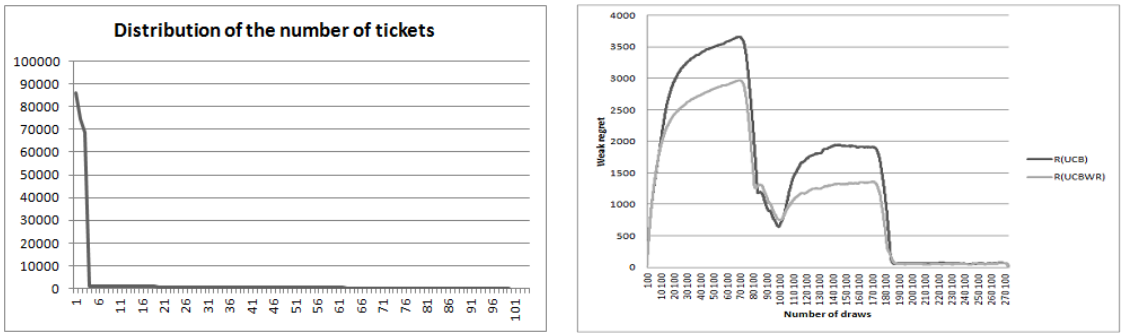


(a) (b)

Figure 2: First synthetic problem. On the left (a), the number of tickets by game. On the right (b), the weak regret: UCB1 outperforms UCBWR when tickets are drawn according to a uniform distribution

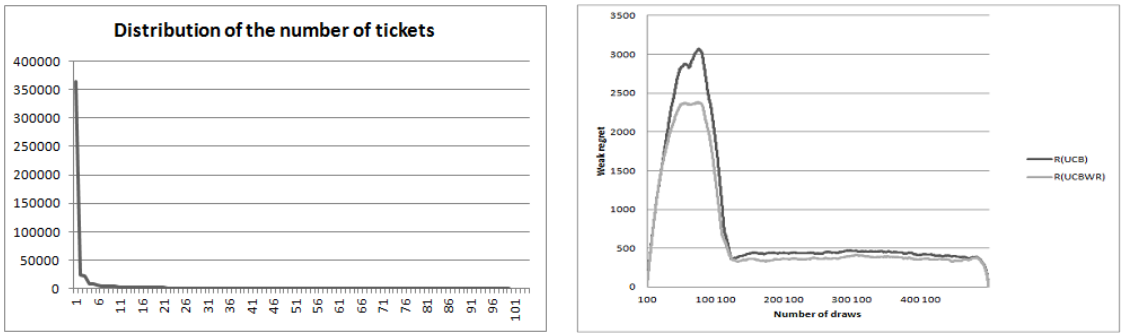
After a small period of times corresponding to the initialization, we observe that the weak regret obtained by UCB1 is lower than the one obtained by UCBWR (see Figure 2b). On the end of the curve, the difference between the two policies is decreasing. It reaches zero when all tickets have been scratched. The numbers of tickets are drawn from the same uniform distribution. In this case, the difference between sampling rates are small and the use of the Serfling inequality rather than the Hoeffding inequality does not give any gain.

In the second synthetic problem, the same games as previously are used, but the number of tickets and the number of winning tickets are multiplied by 100 for games 25, 50 and 75. 273817 tickets including 116374 winning tickets are obtained. In this case, the number of tickets are very different. A few games have a large number of tickets, and lot of games have a similar and small number of tickets (see Figure 3a) . Using the Serfling inequality to evaluate the upper confidence interval, the bounds are very different when the sampling rates are very different. Then UCBWR outperforms UCB1 policy (see Figure 3b).



(a) (b)

Figure 3: Second synthetic problem. On the left (a), the number of tickets by game. On the right (b), the weak regret: UCBWR outperforms UCB1 when the distribution of the number of tickets is far from a uniform distribution

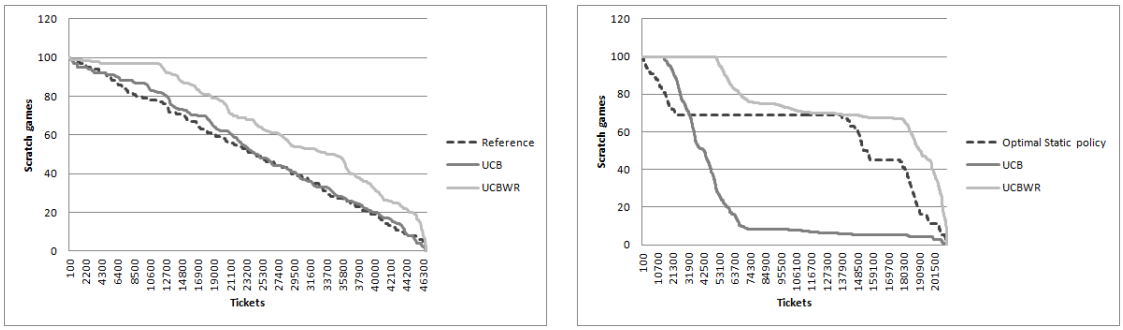


(a) (b)

Figure 4: Third synthetic problem. On the left (a), the number of tickets by game. On the right (b), the weak regret: UCBWR outperforms UCB1 when the distribution of the number of tickets is far from a uniform distribution.

In the last synthetic problem, as in the previous case 100 scratch games with binary reward are used. A pareto distribution is used to draw the number of tickets, with $x_m = 200$ and $\alpha = 1$. In this case, the distribution of the number of tickets is even more unbalanced (see Figure 4a). As in the previous case the numbers of winning tickets are drawn according to a Bernoulli distribution, with p parameter drawn from a uniform distribution between 0 and 0.25. 499524 tickets including 47774 winning tickets spread over 100 scratch games are drawn. The obtained number of tickets and the number of winning tickets are very different for each scratch game. UCBWR outperforms UCB1 policy on the first part of the curve (see Figure 4b).

To deeper investigate the behavior of UCBWR and UCB1 algorithms for scratch games, we have plotted the number of remaining games versus the number of scratched tickets (see Figure 5). In both synthetic problems, we observe that UCBWR tends to keep games longer than UCB1. This is due to the use of the Serfling bound: the exploration term decreases



(a) (b)

Figure 5: Number of remaining games. On the left (a), the numbers of tickets are drawn according to an uniform distribution (first synthetic problem). On the right (b), number of tickets are drawn according to a pareto distribution (third synthetic problem).

faster and the index of the best game tends to be lower than in the case of UCB1. Moreover, due the use of Serfling bound, UCBWR tends to explore more the games with high potential remaining reward, which are the games with high remaining number of tickets. When the numbers of tickets are drawn according to an uniform distribution, the curves of the number of remaining games are similar for the optimal static policy and for UCB1. In this case, the difference between the remaining number of tickets is low and UCB1 outperforms UCBWR. When the numbers of tickets are drawn according to a pareto distribution, there are a few number of games with a lot of tickets. The curves of the number of remaining games are tight for the optimal static policy and for UCBWR. In this case, UCB1 spends to much time to explore and exploit small games and UCBWR outperforms UCB1.

5.3. Test on an Ad Serving application

In the pay-per-click model, the ad server sends different ads on different contexts (profiles $\times$ web pages) in order to maximize the click-through rate. To evaluate the impact of the use of UCBWR policy rather than UCB policy on the ad server optimization, we perform a simulation using the following method:

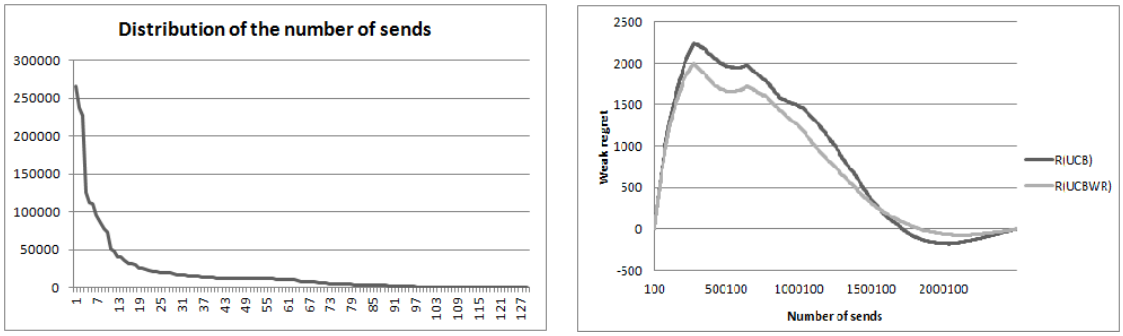
- We have collected for a given web page the number of sends and the number of clicks on each ad during ten days for a sample of 1/1000 of users.
- Each ad is considered as a scratch game, with a finite number of tickets corresponding to the sends of ads, and a finite number of winning tickets corresponding to users who are likely to the click on banners.
- In the simulation, we consider that the observed sends of each ad represent their total inventories on all web pages,

Problem	Number of tickets	Number of winning tickets	$\overline{R}(UCB1)$	$\overline{R}(UCBWR)$
Ad Serving	2500000	11329	870	770
Emailing	11000000	22534	2004	1137

Table 2: Summary of results on the real-world problems

- In order to maximize CTR, the ad server tries to find the best ads for a given web page, and we observe the result on the considered web page when 10% of ads have been displayed.

For the selected web page, the number of ads is on the order of one hundred, the total number of sends is on the order of two and an half millions and the click-through rate is on the order of 0.5 per 1000. Banners and advertisers are very different.



(a) (b)

Figure 6: Ad Serving application. On the left (a), the mumber of sends by ad. On the right (b), the weak regret: UCBWR outperforms UCB1 for this Ad Serving problem.

First, we plot the mean gain of optimal static and optimal policies (see Figure 7). These policies use the knowledge of the mean reward of each scratch game. As expected, these policies provide similar results and as expected these results outperform the ones of UCBWR and UCB1 policies, which do not use the knowledge of mean rewards. Second, the variability of number of tickets is large (see Figure 6a). As expected from the theoretical framework, and as observed on synthetic problems for non-uniform distributions of number of tickets, UCBWR outperforms UCB1 policy (see Figure 6b and Table 2). The obtained results for UCBWR are interesting for a real applications: using this policy on this web page the expected number of clicks is 1.6 times higher than using a random policy (see Figure 7). It corresponds to an increase of incomes of 60%.

5.4. Test on an emailing application

For a telecommunication operator, emailing campaigns are used to promote web services such as self-care, online invoice, and recommendation of movies... In order to respect user's

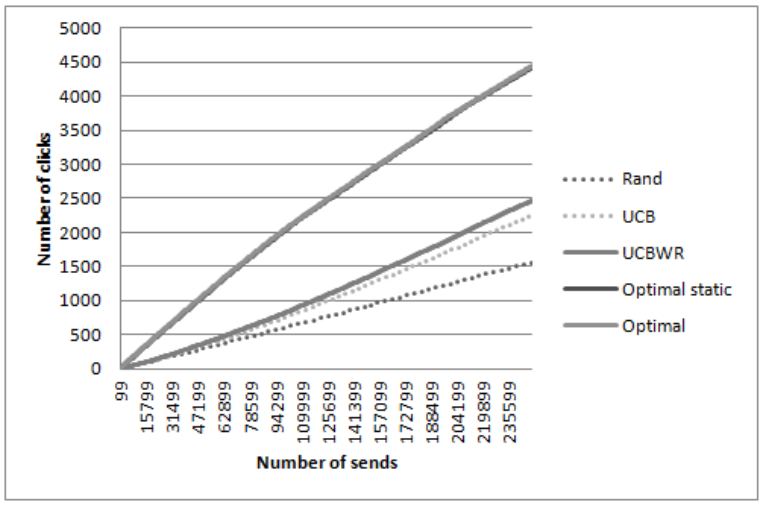
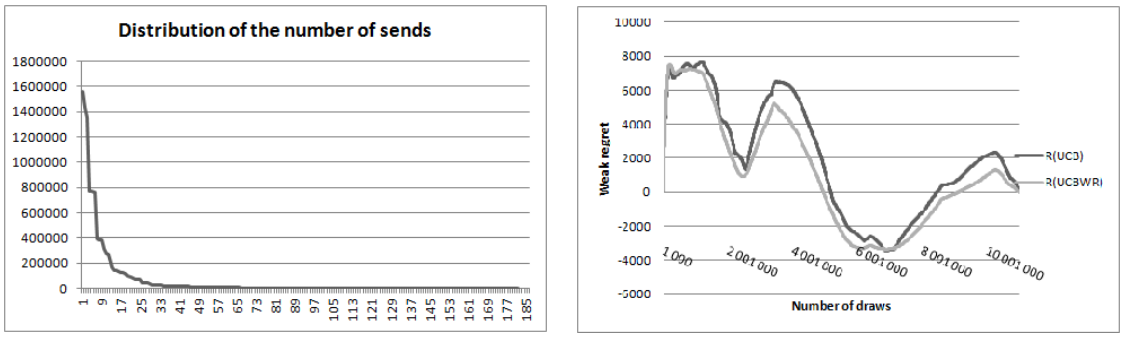


Figure 7: Ad Serving application. The gain of UCB1, UCBWR, optimal static and optimal policies: optimal static and optimal policies are closed and UCBWR outperforms UCB1.

weariness in face of email solicitations, we would like to limit the number of sent emails while maximizing the number of subscriptions. An emailing campaign corresponds to a target, a list of emails selected using their profiles, and a product or service, that we want to promote. The number of emails sent is the inventory of the campaign. The reward of an emailing campaign is the number of clicks on the joint link. In order to maximize the total reward of all emailing campaigns during a time period, we consider for each emailing campaign a potential inventory greater than its planned inventory. The sum of the potential inventories exceeds the planned inventory. The goal of the optimization algorithm is to increase the inventory of emailing campaigns with high reward, and to decrease the inventory of emailing campaigns with low reward, in order to maximize the total reward without exceeding the planned inventory.

We consider that each emailing campaign is a scratch game, with a number of tickets equal to its potential inventory. In order to simulate this emailing application, we have collected 182 emailing campaigns corresponding to eleven millions of sent emails and 220000 clicks on the links of proposed services. First, we confirm that optimal static and optimal policies provide similar results (see Figure 8b). Second, for this non-uniform distribution of the number of tickets (see Figure 8a), UCBWR outperforms UCB1 (see Figure 8b, Table 2) . The result of the optimization allows to have the same impact while reducing the user's weariness in face of email solicitations. For example, to obtain 60000 clicks on the enclosed link, using UCBWR policy 1650000 emails sent are needed, and using a random policy 2150000 emails sent are needed (Figure 9). It corresponds to a reduction of 23% of the sent emails.



(a) (b)

Figure 8: Optimization of emailing campaigns. On the left (a), the number of sends by campaign. On the right (b), the weak regret: UCBWR outperforms UCB1 on optimization of emailing campaigns.

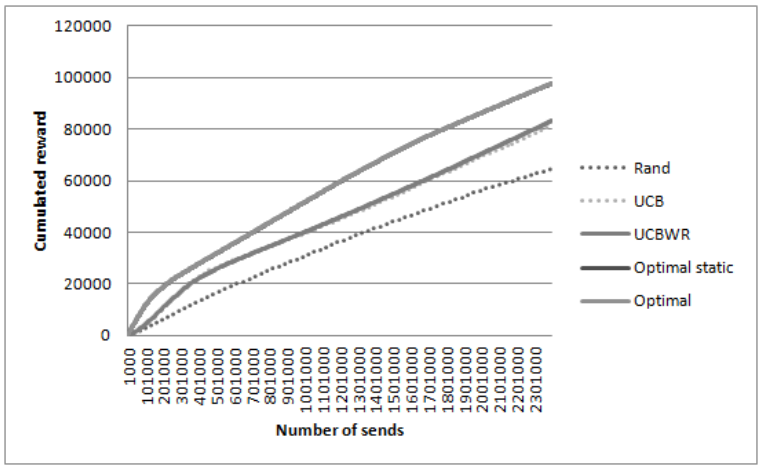


Figure 9: Optimization of emailing campaigns. The gain of UCB1, UCBWR, optimal static and optimal policies: optimal static and optimal policies are tight and UCBWR outperforms UCB1.

6. Conclusion

We have proposed a new problem setup, the scratch games, to take into account finite sequence of rewards in bandit algorithms. We redefined the weak regret for scratch games. We have proposed a new upper confidence bound policy based on Serfling inequality rather than Chernoff-Hoeffding inequality. We have shown that the bound of the expectation of playing a suboptimal scratch game using UCBWR is lower than the one using UCB1 policy. We have provided a bound of this expectation. We studied the behavior of UCBWR algorithm for different synthetic problems to complete theoretical results. Our experiments confirm that UCBWR outperforms UCB1 policy when the distribution of the number of tickets is far from an uniform distribution. This setting corresponds to two real world problems: selecting ads to display on web pages and optimizing emailing campaigns.

When the sequence of rewards is finite, the proposed upper confidence bound based on Serfling inequality can be used for all the policies which are derived from UCB policy, and in particular UCT algorithm Kocsis and Szeoesvàri (2006). Indeed, in the case of tree structured data, usually the number of elements of each node is very unbalanced. As our experiments have shown, we can expect a gain using Serfling inequality rather than Chernoff-Hoeffding inequality.

References

Rajeev Agrawal. Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27:1054–1078, 1995.

Jean-Yves Audibert, Rémi Munos, and Csaba Szeoesvàri. Exploration-exploitation tradeoff using variance estiumates in multi-armed bandits. *Theoretical Computer Science*, 410 (19):1876–1902, 2009.

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

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvàri. Online Optimization in X-Armed Bandits. In *Neural Information Processing Systems*, Vancouver, Canada, 2008.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *NIPS*, pages 273–280, 2008.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *COLT*, 2011.

Emilie Kaufman, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandits problems. In *AISTATS*, 2012.

Levente Kocsis and Csaba Szeoesvàri. Bandit based monte-carlo planning. In *ECML*, pages 282–293, 2006.

T. L. Lai and Herbert Robbins. Asymptotivally efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

Sandeep Pandey, Deepak Agarwal, and Deepayan Chakrabarti. Multi-armed bandit problems with dependent arms. In *ICML*, pages 721–728, 2007a.

Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: A model-based approach. In *SIAM*, pages 216–227, 2007b.

R. J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2:39–48, 1974.

Sham M. Kakade Varsha Dani, Thomas P. Hayes. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.