Garbage In, Reward Out: Bootstrapping Exploration in Multi-Armed Bandits

Branislav Kveton [1] **Csaba Szepesvári** [2 3] **Sharan Vaswani** [4] **Zheng Wen** [5] **Mohammad Ghavamzadeh** [6]
Tor Lattimore [2]

Abstract

We propose a bandit algorithm that explores by randomizing its history of rewards. Specifically, it pulls the arm with the highest mean reward in a non-parametric bootstrap sample of its history with pseudo rewards. We design the pseudo rewards such that the bootstrap mean is optimistic with a sufficiently high probability. We call our algorithm `Giro`, which stands for *garbage in, reward out*. We analyze `Giro` in a Bernoulli bandit and derive a $O(K\Delta^{-1}\log n)$ bound on its n-round regret, where Δ is the difference in the expected rewards of the optimal and the best suboptimal arms, and K is the number of arms. The main advantage of our exploration design is that it easily generalizes to structured problems. To show this, we propose contextual `Giro` with an arbitrary reward generalization model. We evaluate `Giro` and its contextual variant on multiple synthetic and real-world problems, and observe that it performs well.

1. Introduction

A *multi-armed bandit* (Lai and Robbins, 1985; Auer et al., 2002; Lattimore and Szepesvari, 2019) is an online learning problem where actions of the *learning agent* are represented by *arms*. The arms can be treatments in a clinical trial or ads on a website. After the arm is *pulled*, the agent receives its *stochastic reward*. The objective of the agent is to maximize its expected cumulative reward. The agent does not know the expected rewards of the arms and thus faces the so-called *exploration-exploitation dilemma*: *explore*, and learn more about arms; or *exploit*, and pull the arm with the highest estimated reward thus far.

A *contextual bandit* (Li et al., 2010; Agrawal and Goyal, 2013b) is a generalization of a multi-armed bandit where the learning agent has access to additional context in each round. The context can encode the medical data of a patient in a clinical trial or the demographic information of a targeted user on a website. In this case, the expected reward is an unknown function of the arm and context. This function is often parametric and its parameters are learned. In *linear bandits* (Rusmevichientong and Tsitsiklis, 2010; Dani et al., 2008; Abbasi-Yadkori et al., 2011), this function is linear; and the expected reward is the dot product of a known context vector and an unknown parameter vector.

Arguably, the most used and studied exploration strategies in multi-armed and contextual bandits are *Thompson sampling* (Thompson, 1933; Agrawal and Goyal, 2013a), the *optimism in the face of uncertainty* (Auer et al., 2002; Abbasi-Yadkori et al., 2011), and the *ϵ-greedy policy* (Sutton and Barto, 1998; Auer et al., 2002). The *ϵ-greedy policy* is general and thus widely used in practice. However, it is also statistically suboptimal. Its performance heavily depends on the value of ϵ and the strategy for annealing it.

Optimism in the face of uncertainty (OFU) relies on high-probability confidence sets. These sets are statistically and computationally efficient in multi-armed and linear bandits (Auer et al., 2002; Abbasi-Yadkori et al., 2011). However, when the reward function is non-linear in context, we only know how to construct approximate confidence sets (Filippi et al., 2010; Zhang et al., 2016; Li et al., 2017; Jun et al., 2017). These sets tend to be conservative (Filippi et al., 2010) and statistically suboptimal.

The key idea in *Thompson sampling (TS)* is to maintain a posterior distribution over model parameters and then act optimistically with respect to samples from it. TS is computationally efficient when the posterior distribution has a closed form, as in multi-armed bandits with Bernoulli and Gaussian rewards. If the posterior does not have a closed form, it has to be approximated. Computationally efficient approximations exist in multi-armed bandits with $[0, 1]$ rewards (Agrawal and Goyal, 2013a). Such approximations are costly in general (Gopalan et al., 2014; Kawale et al., 2015; Lu and Van Roy, 2017; Riquelme et al., 2018).

To address these problems, bootstrapping exploration has been proposed in both multi-armed and contextual bandits

[1]Google Research [2]DeepMind [3]University of Alberta [4]Mila, University of Montreal [5]Adobe Research [6]Facebook AI Research. Correspondence to: Branislav Kveton <bkveton@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

(Baransi et al., 2014; Eckles and Kaptein, 2014; Osband and Van Roy, 2015; Tang et al., 2015; Elmachtoub et al., 2017; Vaswani et al., 2018). Bootstrapping has two advantages over existing exploration strategies. First, unlike OFU and TS, it is easy to implement in any problem, because it does not require problem-specific confidence sets or posteriors. Second, unlike the ϵ-greedy policy, it is data driven and not sensitive to tuning. Despite its advantages and good empirical performance, exploration by bootstrapping is poorly understood theoretically. The strongest theoretical result is that of Osband and Van Roy (2015), who showed that a form of bootstrapping in a Bernoulli bandit is equivalent to Thompson sampling.

We make the following contributions in this paper. First, we propose a general randomized algorithm that explores conditioned on its history. We show that some instances of this algorithm are not sound. Second, we propose `Giro`, an algorithm that pulls the arm with the highest mean reward in a non-parametric bootstrap sample of its history with pseudo rewards. We design the pseudo rewards such that the bootstrap mean is optimistic with a high probability. Third, we analyze `Giro` in a K-armed Bernoulli bandit and prove a $O(K\Delta^{-1}\log n)$ bound on its n-round regret, where Δ is the difference in the expected rewards of the optimal and the best suboptimal arms. Our analyses of the general randomized algorithm and `Giro` provide novel insights on how randomness helps with exploration. Fourth, we propose contextual `Giro`. Finally, we empirically evaluate `Giro` and its contextual variant on both synthetic and real-world problems, and observe good performance.

2. Setting

We adopt the following notation. The set $\{1, \dots, n\}$ is denoted by $[n]$. We define $\mathrm{Ber}(x; p) = p^x(1-p)^{1-x}$ and let $\mathrm{Ber}(p)$ be the corresponding Bernoulli distribution. In addition, we define $B(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x}$ and let $B(n, p)$ be the corresponding binomial distribution. For any event E, $\mathbb{1}\{E\} = 1$ if event E occurs and $\mathbb{1}\{E\} = 0$ otherwise. We introduce multi-armed and contextual bandits below.

A *multi-armed bandit* (Lai and Robbins, 1985; Auer et al., 2002; Lattimore and Szepesvari, 2019) is an online learning problem where the learning agent pulls K arms in n rounds. In round $t \in [n]$, the agent pulls arm $I_t \in [K]$ and receives its reward. The reward of arm $i \in [K]$ in round t, $Y_{i,t}$, is drawn i.i.d. from a distribution of arm i, P_i, with mean μ_i and support $[0, 1]$. The means are unknown to the learning agent. The objective of the agent is to maximize its expected cumulative reward in n rounds.

We assume that arm 1 is optimal, that is $\mu_1 > \max_{i>1} \mu_i$. Let $\Delta_i = \mu_1 - \mu_i$ be the *gap* of arm i, the difference in the expected rewards of arms 1 and i. Then maximization of the expected n-round reward can be viewed as minimizing the *expected n-round regret*, which we define as

$$R(n) = \sum_{i=2}^{K} \Delta_i \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}\{I_t = i\}\right]. \tag{1}$$

A *contextual bandit* (Li et al., 2010; Agrawal and Goyal, 2013b) is a generalization of a multi-armed bandit where the learning agent observes context $x_t \in \mathbb{R}^d$ at the beginning of each round t. The reward of arm i in round t is drawn i.i.d. from a distribution that depends on both arm i and x_t. One example is a logistic reward model, where

$$Y_{i,t} \sim \mathrm{Ber}(1/(1 + \exp[-x_t^\intercal \theta_i])) \tag{2}$$

and $\theta_i \in \mathbb{R}^d$ is the parameter vector associated with arm i. The learning agent does not know θ_i.

If the reward $Y_{i,t}$ was generated as in (2), we could solve our contextual bandit problem as a generalized linear bandit (Filippi et al., 2010). However, if $Y_{i,t}$ was generated by a more complex function of context x_t, such as a neural network, we would not know how to design a sound bandit algorithm. The difficulty is not in modeling uncertainty; it is in the lack of computationally efficient methods to do it. For instance, Thompson sampling can be analyzed in very general settings (Gopalan et al., 2014). However, as discussed in Section 1, accurate posterior approximations are computationally expensive.

3. General Randomized Exploration

In this section, we present a general randomized algorithm that explores conditioned on its history and context. Later, in Sections 4 and 5, we propose and analyze an instance of this algorithm with sublinear regret.

We use the following notation. The *history* of arm i after s pulls is a vector $\mathcal{H}_{i,s}$ of length s. The j-th entry of $\mathcal{H}_{i,s}$ is a pair $(x_\ell, Y_{i,\ell})$, where ℓ is the index of the round where arm i is pulled for the j-th time. We define $\mathcal{H}_{i,0} = ()$. The *number of pulls* of arm i in the first t rounds is denoted by $T_{i,t}$ and defined as $T_{i,t} = \sum_{\ell=1}^{t} \mathbb{1}\{I_\ell = i\}$.

Our meta algorithm is presented in Algorithm 1. In round t, the algorithm draws the value of each arm i, $\hat{\mu}_{i,t}$, from distribution p (line 5), which depends on the history of the arm $\mathcal{H}_{i,s}$ and the context x_t in round t. The arm with the highest value is pulled (line 6), and its history is extended by a pair of context x_t and the reward of arm I_t (line 11). We denote by $u \oplus v$ the concatenation of vectors u and v.

We present a general contextual algorithm because we return to it in Sections 4.3 and 6.2. For now, we restrict our attention to multi-armed bandits. To simplify exposition, we omit context from the entries of $\mathcal{H}_{i,s}$.

Algorithm 1 General randomized exploration.

1: $\forall i \in [K] : T_{i,0} \leftarrow 0,\ \mathcal{H}_{i,0} \leftarrow ()$ ▷ Initialization
2: **for** $t = 1, \dots, n$ **do**
3: **for** $i = 1, \dots, K$ **do** ▷ Estimate arm values
4: $s \leftarrow T_{i,t-1}$
5: Draw $\hat{\mu}_{i,t} \sim p(\mathcal{H}_{i,s}, x_t)$
6: $I_t \leftarrow \arg\max_{i \in [K]} \hat{\mu}_{i,t}$ ▷ Pulled arm
7: Pull arm I_t and observe $Y_{i,t}$
8: **for** $i = 1, \dots, K$ **do** ▷ Update statistics
9: **if** $i = I_t$ **then**
10: $T_{i,t} \leftarrow T_{i,t-1} + 1$
11: $\mathcal{H}_{i,T_{i,t}} \leftarrow \mathcal{H}_{i,T_{i,t-1}} \oplus ((x_t, Y_{i,t}))$
12: **else**
13: $T_{i,t} \leftarrow T_{i,t-1}$

Algorithm 1 is instantiated by designing the distribution p in line 5. This distribution can be designed in many ways. In Bernoulli TS (Agrawal and Goyal, 2013a), for instance, p is a beta posterior distribution. A less direct approach to designing p is to resample the history of rewards in each round, as in bootstrapping exploration (Section 7). More specifically, let $\mathcal{B}_{i,s}$ be a *non-parametric bootstrap sample* (Efron and Tibshirani, 1986) of arm i after s pulls. We let $\mathcal{B}_{i,s}$ be a vector of the same length as $\mathcal{H}_{i,s}$ and assume that its entries are drawn with replacement from the entries of $\mathcal{H}_{i,s}$ in each round. Then the value of arm i in round t is estimated as

$$\hat{\mu}_{i,t} = \frac{1}{|\mathcal{B}_{i,s}|} \sum_{y \in \mathcal{B}_{i,s}} y\,, \tag{3}$$

where $s = T_{i,t-1}$. Note that we slightly abuse our notation and treat vectors as sets. In the next section, we show that this natural instance of Algorithm 1 can have linear regret.

3.1. Linear Regret

The variant of Algorithm 1 in (3) can have linear regret in a Bernoulli bandit with $K = 2$ arms. More specifically, if the first reward of the optimal arm is 0, its estimated value remains 0 until the arm is pulled again, which may never happen if the estimated value of the other arm is positive. We formally state this result below.

Without loss of generality, let $\mu_1 > \mu_2$. Algorithm 1 is implemented as follows. Both arms are initially pulled once, arm 1 in round 1 and arm 2 in round 2. The value of arm i in round t is computed as in (3). If $\hat{\mu}_{1,t} = \hat{\mu}_{2,t}$, the tie is broken by a fixed rule that is chosen randomly in advance, as is common in multi-armed bandits. In particular, Algorithm 1 draws $Z \sim \mathrm{Ber}(1/2)$ before the start of round 1. If $\hat{\mu}_{1,t} = \hat{\mu}_{2,t}$, $I_t = \mathbb{1}\{Z = 1\} + 1$. We bound the regret of this algorithm below.

Lemma 1. *In a Bernoulli bandit with 2 arms, the expected n-round regret of the above variant of Algorithm 1 can be bounded from below as $R(n) \geq 0.5\,(1 - \mu_1)\Delta_2(n-1)$.*

Proof. By design, the algorithm does not pull arm 1 after event $E = \{Z = 1,\ \mathcal{H}_{1,1} = (0)\}$ occurs. Since the events $Z = 1$ and $\mathcal{H}_{1,1} = (0)$ are independent,

$$\mathbb{P}\left(E\right) = \mathbb{P}\left(Z = 1\right) \mathbb{P}\left(\mathcal{H}_{1,1} = (0)\right) = 0.5\,(1 - \mu_1)\,.$$

Moreover, if event $\mathcal{H}_{1,1} = (0)$ occurs, it must occur by the end of round 1 because the algorithm pulls arm 1 in round 1. Now we combine the above two facts and get

$$\begin{aligned} R(n) &\geq \mathbb{E}\left[\left(\sum_{t=2}^{n} \Delta_2 \mathbb{1}\{I_t = 2\}\right) \mathbb{1}\{E\}\right] \\ &= \mathbb{E}\left[\sum_{t=2}^{n} \Delta_2 \mathbb{1}\{I_t = 2\} \,\middle|\, E\right] \mathbb{P}\left(E\right) \\ &= 0.5\,(1 - \mu_1)\Delta_2(n-1)\,. \end{aligned}$$

This concludes the proof. ■

A similar lower bound, $R(n) \geq 0.5\,(1 - \mu_1)^k \Delta_2(n-k)$, can be derived in the setting where each arm is initialized by k pulls. This form of forced exploration was proposed in bootstrapping bandits earlier (Tang et al., 2015; Elmachtoub et al., 2017), and is clearly not sound. A similar argument to Lemma 1, although less formal, is in Section 3.1 of Osband and Van Roy (2015).

4. Garbage In, Reward Out

One solution to the issues in Section 3.1 is to add positive and negative pseudo rewards, 1 and 0, to $\mathcal{H}_{i,s}$. This increases the variance of the bootstrap mean in (3) and may lead to exploration. However, the pseudo rewards also introduce bias that has to be controlled. In the next section, we present a design that trades off these two quantities.

4.1. Algorithm `Giro`

We propose Algorithm 2, which increases the variance of the bootstrap mean in (3) by adding pseudo rewards to the history of the pulled arm (line 18). The algorithm is called `Giro`, which stands for *garbage in, reward out*. This is an informal description of our exploration strategy, which adds seemingly useless extreme rewards to the history of the pulled arm. We call them *pseudo rewards*, to distinguish them from observed rewards. `Giro` has one tunable parameter a, the number of positive and negative pseudo rewards in the history for each observed reward.

`Giro` does not seem sound, because the number of pseudo rewards in history $\mathcal{H}_{i,s}$ grows linearly with s. In fact, this

is the key idea in our design. We justify it informally in the next section and bound its regret in Section 5.

4.2. Informal Justification

In this section, we informally justify the design of `Giro` in a Bernoulli bandit. Our argument has two parts. First, we show that $\hat{\mu}_{i,t}$ in line 8 *concentrates* at the scaled and shifted expected reward of arm i, which preserves the order of the arms. Second, we show that $\hat{\mu}_{i,t}$ is *optimistic*, its value is higher than the scaled and shifted expected reward of arm i, with a sufficiently high probability. This is sufficient for the regret analysis in Section 5.

Fix arm i and the number of its pulls s. Let $V_{i,s}$ denote the number of ones in history $\mathcal{H}_{i,s}$, which includes a positive and negative pseudo rewards for each observed reward of arm i. By definition, $V_{i,s} - as$ is the number of positive observed rewards of arm i. These rewards are drawn i.i.d. from $\mathrm{Ber}(\mu_i)$. Thus $V_{i,s} - as \sim B(s, \mu_i)$ and we have

$$\mathbb{E}\left[V_{i,s}\right] = (\mu_i + a)s\,, \quad \mathrm{var}\left[V_{i,s}\right] = \mu_i(1-\mu_i)s\,. \tag{4}$$

Now we define $\alpha = 2a + 1$ and let $U_{i,s}$ be the number of ones in bootstrap sample $\mathcal{B}_{i,s}$. Since drawing αs samples with replacement from $\mathcal{H}_{i,s}$ is equivalent to drawing αs i.i.d. samples from $\mathrm{Ber}(V_{i,s}/(\alpha s))$, we have $U_{i,s} \mid V_{i,s} \sim B(\alpha s, V_{i,s}/(\alpha s))$. From the definition of $U_{i,s}$, we have for any $V_{i,s}$,

$$\mathbb{E}\left[U_{i,s} \mid V_{i,s}\right] = V_{i,s}\,, \tag{5}$$

$$\mathrm{var}\left[U_{i,s} \mid V_{i,s}\right] = V_{i,s}\left(1 - \frac{V_{i,s}}{\alpha s}\right)\,. \tag{6}$$

Let $\hat{\mu} = U_{i,s}/(\alpha s)$ be the mean reward in $\mathcal{B}_{i,s}$. First, we argue that $\hat{\mu}$ concentrates. From the properties of $U_{i,s}$ in (5) and (6), for any $V_{i,s}$, we have

$$\mathbb{E}\left[\hat{\mu} \mid V_{i,s}\right] = \frac{V_{i,s}}{\alpha s}\,, \quad \mathrm{var}\left[\hat{\mu} \mid V_{i,s}\right] = \frac{V_{i,s}}{\alpha^2 s^2}\left(1 - \frac{V_{i,s}}{\alpha s}\right)\,.$$

Since $V_{i,s} \in [as, (a+1)s]$, $\mathrm{var}\left[\hat{\mu} \mid V_{i,s}\right] = O(1/s)$ and $\hat{\mu} \to V_{i,s}/(\alpha s)$ as s increases. Moreover, from the properties of $V_{i,s}$ in (4), we have

$$\mathbb{E}\left[\frac{V_{i,s}}{\alpha s}\right] = \frac{\mu_i + a}{\alpha}\,, \quad \mathrm{var}\left[\frac{V_{i,s}}{\alpha s}\right] = \frac{\mu_i(1-\mu_i)}{\alpha^2 s}\,.$$

So, $V_{i,s}/(\alpha s) \to (\mu_i + a)/\alpha$ as s increases. By transitivity, $\hat{\mu} \to (\mu_i + a)/\alpha$, which is the scaled and shifted expected reward of arm i. This transformation preserves the order of the arms; but it changes the gaps, the differences in the expected rewards of the optimal and suboptimal arms.

Second, we argue that $\hat{\mu}$ is *optimistic*, that any unfavorable history is less likely than being optimistic under that history. In particular, let $E = \{V_{i,s}/(\alpha s) = (\mu_i + a)/\alpha - \varepsilon\}$ be the event that the mean reward in the history deviates from its expectation by $\varepsilon > 0$. Then

$$\mathbb{P}\left(\hat{\mu} \geq (\mu_i + a)/\alpha \mid E\right) \geq \mathbb{P}\left(E\right) \tag{7}$$

holds for any $\varepsilon > 0$ such that $\mathbb{P}\left(E\right) > 0$.

Trivially, $\mathbb{P}\left(E\right) \leq \mathbb{P}\left(V_{i,s}/(\alpha s) \leq (\mu_i + a)/\alpha - \varepsilon\right)$ holds for any $\varepsilon > 0$. Therefore, if both $V_{i,s}/(\alpha s)$ and $\hat{\mu} \mid V_{i,s}$ were normally distributed, inequality (7) would hold if for any $V_{i,s}$, $\mathrm{var}\left[\hat{\mu} \mid V_{i,s}\right] \geq \mathrm{var}\left[V_{i,s}/(\alpha s)\right]$. We compare the variances below. Since $V_{i,s} \in [as, (a+1)s]$,

$$\begin{aligned}\mathrm{var}\left[\hat{\mu} \mid V_{i,s}\right] &\geq \frac{1}{\alpha s}\min_{v \in [as,(a+1)s]} \frac{v}{\alpha s}\left(1 - \frac{v}{\alpha s}\right) \\ &= \frac{1}{\alpha s}\frac{as}{\alpha s}\left(1 - \frac{as}{\alpha s}\right) = \frac{a(a+1)}{\alpha^3 s}\,.\end{aligned}$$

Trivially, $\mathrm{var}\left[V_{i,s}/(\alpha s)\right] \leq 1/(4\alpha^2 s)$. Therefore, for any a such that $a(a+1)/\alpha \geq 1/4$, roughly $a \geq 1/3$, $\hat{\mu}$ is optimistic. We formalize this intuition in Section 5. A formal proof is necessary because our assumption of normality was unrealistic.

Algorithm 2 `Giro` with $[0, 1]$ rewards.

1: **Inputs:** Pseudo rewards per unit of history a
2: $\forall i \in [K] : T_{i,0} \leftarrow 0,\ \mathcal{H}_{i,0} \leftarrow ()$ ▷ Initialization
3: **for** $t = 1, \ldots, n$ **do**
4: **for** $i = 1, \ldots, K$ **do** ▷ Estimate arm values
5: **if** $T_{i,t-1} > 0$ **then**
6: $s \leftarrow T_{i,t-1}$
7: $\mathcal{B}_{i,s} \leftarrow$ Sample $|\mathcal{H}_{i,s}|$ times from $\mathcal{H}_{i,s}$ with replacement
8: $\hat{\mu}_{i,t} \leftarrow \dfrac{1}{|\mathcal{B}_{i,s}|}\displaystyle\sum_{y \in \mathcal{B}_{i,s}} y$
9: **else**
10: $\hat{\mu}_{i,t} \leftarrow +\infty$
11: $I_t \leftarrow \arg\max_{i \in [K]} \hat{\mu}_{i,t}$ ▷ Pulled arm
12: Pull arm I_t and get reward $Y_{I_t,t}$
13: **for** $i = 1, \ldots, K$ **do** ▷ Update statistics
14: **if** $i = I_t$ **then**
15: $T_{i,t} \leftarrow T_{i,t-1} + 1$
16: $\mathcal{H}_{i,T_{i,t}} \leftarrow \mathcal{H}_{i,T_{i,t-1}} \oplus (Y_{i,t})$
17: **for** $\ell = 1, \ldots, a$ **do** ▷ Pseudo rewards
18: $\mathcal{H}_{i,T_{i,t}} \leftarrow \mathcal{H}_{i,T_{i,t}} \oplus (0, 1)$
19: **else**
20: $T_{i,t} \leftarrow T_{i,t-1}$

4.3. Contextual `Giro`

We generalize `Giro` to a contextual bandit in Algorithm 3. The main difference in Algorithm 3 is that it fits a reward generalization model to $\mathcal{B}_{i,s}$ and then estimates the value

Algorithm 3 Contextual `Giro` with $[0, 1]$ rewards.

1: **Inputs:** Pseudo rewards per unit of history a
2: $\forall i \in [K] : T_{i,0} \leftarrow 0,\ \mathcal{H}_{i,0} \leftarrow ()$ ▷ Initialization
3: **for** $t = 1, \dots, n$ **do**
4: **for** $i = 1, \dots, K$ **do** ▷ Estimate arm values
5: **if** $T_{i,t-1} > 0$ **then**
6: $s \leftarrow T_{i,t-1}$
7: $\mathcal{B}_{i,s} \leftarrow$ Sample $|\mathcal{H}_{i,s}|$ times from $\mathcal{H}_{i,s}$ with replacement
8: $\hat{\mu}_{i,t} \leftarrow$ Estimate $\mathbb{E}\left[Y_{i,t} \mid \mathcal{B}_{i,s}, x_t\right]$
9: **else**
10: $\hat{\mu}_{i,t} \leftarrow +\infty$
11: $I_t \leftarrow \arg\max_{i \in [K]} \hat{\mu}_{i,t}$ ▷ Pulled arm
12: Pull arm I_t and get reward $Y_{I_t,t}$
13: **for** $i = 1, \dots, K$ **do** ▷ Update statistics
14: **if** $i = I_t$ **then**
15: $T_{i,t} \leftarrow T_{i,t-1} + 1$
16: $\mathcal{H}_{i,T_{i,t}} \leftarrow \mathcal{H}_{i,T_{i,t-1}} \oplus ((x_t, Y_{i,t}))$
17: **for** $\ell = 1, \dots, a$ **do** ▷ Pseudo rewards
18: $\mathcal{H}_{i,T_{i,t}} \leftarrow \mathcal{H}_{i,T_{i,t}} \oplus ((x_t, 0), (x_t, 1))$
19: **else**
20: $T_{i,t} \leftarrow T_{i,t-1}$

of arm i in context x_t (line 8) based on this model. If this model was linear with parameters θ_i, the estimated value would be $x_t^\intercal \theta_i$. The other difference is that the pseudo rewards are associated with context x_t (line 18), to increase the conditional variance of the estimates given x_t.

The value of arm i in round t, $\mathbb{E}\left[Y_{i,t} \mid \mathcal{B}_{i,s}, x_t\right]$, can be approximated by any function of x_t that can be learned from $\mathcal{B}_{i,s}$. The approximation should permit any constant shift of any representable function. In linear models, this can be achieved by adding a bias term to x_t. We experiment with multiple reward generalization models in Section 6.2.

5. Analysis

In Section 5.1, we prove an upper bound on the expected n-round regret of Algorithm 1. In Section 5.2, we prove an upper bound on the expected n-round regret of `Giro` in a Bernoulli bandit. In Section 5.3, we discuss the results of our analysis. Note that our analysis is in the multi-armed bandit setting.

5.1. General Randomized Exploration

We prove an upper bound on the regret of Algorithm 1 in a multi-armed bandit with K arms below. The setting and regret are formally defined in Section 2. The distribution p in line 5 of Algorithm 1 is a function of the history of the arm. For $s \in [n] \cup \{0\}$, let

$$Q_{i,s}(\tau) = \mathbb{P}\left(\hat{\mu} \geq \tau \mid \hat{\mu} \sim p(\mathcal{H}_{i,s}),\ \mathcal{H}_{i,s}\right) \tag{8}$$

be the tail probability that $\hat{\mu}$ conditioned on history $\mathcal{H}_{i,s}$ is at least τ for some tunable parameter τ.

Theorem 1. *For any tunable parameters $(\tau_i)_{i=2}^K \in \mathbb{R}^{K-1}$, the expected n-round regret of Algorithm 1 can be bounded from above as $R(n) \leq \sum_{i=2}^K \Delta_i (a_i + b_i)$, where*

$$a_i = \sum_{s=0}^{n-1} \mathbb{E}\left[\min\left\{1/Q_{1,s}(\tau_i) - 1, n\right\}\right],$$
$$b_i = \sum_{s=0}^{n-1} \mathbb{P}\left(Q_{i,s}(\tau_i) > 1/n\right) + 1.$$

Proof. A detailed proof is in Appendix A. It is motivated by the proof of Thompson sampling (Agrawal and Goyal, 2013a). Our main contribution is that we state and prove the claim such that it can be reused for the regret analysis of any sampling distribution in Algorithm 1. ■

Theorem 1 says that the regret of Algorithm 1 is low when $Q_{i,s}(\tau_i) \to 0$ and $Q_{1,s}(\tau_i) \to 1$ as $s \to \infty$. This suggests the following setting of the tunable parameter τ_i for $i > 1$. When $p(\mathcal{H}_{i,s})$ concentrates at μ_i and $\mu_i < \mu_1$, τ_i should be chosen from interval (μ_i, μ_1). Then $Q_{i,s}(\tau_i) \to 0$ and $Q_{1,s}(\tau_i) \to 1$ would follow by concentration as $s \to \infty$. In general, when $p(\mathcal{H}_{i,s})$ concentrates at μ_i' and $\mu_i' < \mu_1'$, τ_i should be chosen from interval (μ_i', μ_1').

5.2. Bernoulli `Giro`

We analyze `Giro` in a K-armed Bernoulli bandit. We make an additional assumption over Section 5.1 that the rewards are binary. Our regret bound is stated below.

Theorem 2. *For any $a > 1/\sqrt{2}$, the expected n-round regret of `Giro` is bounded from above as*

$$R(n) \leq \sum_{i=2}^K \Delta_i \left[\underbrace{\left(\frac{16(2a+1)c}{\Delta_i^2} \log n + 2\right)}_{\text{Upper bound on } a_i \text{ in Theorem 1}} + \underbrace{\left(\frac{8(2a+1)}{\Delta_i^2} \log n + 2\right)}_{\text{Upper bound on } b_i \text{ in Theorem 1}} \right], \tag{9}$$

where $b = (2a+1)/[a(a+1)]$ and

$$c = \frac{2e^2\sqrt{2a+1}}{\sqrt{2\pi}} \exp\left[\frac{8b}{2-b}\right] \left(1 + \sqrt{\frac{2\pi}{4-2b}}\right)$$

is an upper bound on the expected inverse probability of being optimistic, which is derived in Appendix C.

Proof. The claim is proved in Appendix B. The key steps in the analysis are outlined below. ■

Since `Giro` is an instance of Algorithm 1, we prove Theorem 2 using Theorem 1, where we instantiate distribution specific artifacts. In the notation of Section 5.1, $Q_{i,s}(\tau)$ in (8) is the probability that the mean reward in the bootstrap sample $\mathcal{B}_{i,s}$ of arm i after s pulls is at least τ, conditioned on history $\mathcal{H}_{i,s}$.

Fix any suboptimal arm i. Based on Section 4.2, the mean reward in $\mathcal{B}_{i,s}$ concentrates at $\mu_i' = (\mu_i + a)/\alpha$, where $\alpha = 2a + 1$. Following the discussion on the choice of τ_i in Section 5.1, we set $\tau_i = (\mu_i + a)/\alpha + \Delta_i/(2\alpha)$, which is the average of μ_i' and μ_1'. Recall that $V_{i,s}$ is the number of ones in history $\mathcal{H}_{i,s}$ and $U_{i,s}$ is the number of ones in its bootstrap sample $\mathcal{B}_{i,s}$. Then $Q_{i,s}(\tau_i)$ can be written as

$$Q_{i,s}(\tau_i) = \mathbb{P}\left(\frac{U_{i,s}}{\alpha s} \geq \frac{\mu_i + a}{\alpha} + \frac{\Delta_i}{2\alpha} \,\middle|\, V_{i,s}\right) \text{ for } s > 0\,,$$
$$Q_{i,0}(\tau_i) = 1\,,$$

where $Q_{i,0}(\tau_i) = 1$ because of the initialization in line 10 of `Giro`. We define $Q_{1,s}(\tau_i)$ analogously, by replacing $U_{i,s}$ with $U_{1,s}$ and $V_{i,s}$ with $V_{1,s}$.

The bound in Theorem 2 is proved as follows. To simplify notation, we introduce $a_{i,s} = \mathbb{E}\left[\min\left\{1/Q_{1,s}(\tau_i) - 1, n\right\}\right]$ and $b_{i,s} = \mathbb{P}\left(Q_{i,s}(\tau_i) > 1/n\right)$. By concentration, $a_{i,s}$ and $b_{i,s}$ are small when the number of pulls s is large, on the order of $\Delta_i^{-2}\log n$. When the number of pulls s is small, we bound $b_{i,s}$ trivially by 1 and $a_{i,s}$ as

$$a_{i,s} \leq \mathbb{E}\left[1/\mathbb{P}\left(U_{1,s} \geq (\mu_1 + a)s \,|\, V_{1,s}\right)\right]\,,$$

where the right-hand side is bounded in Appendix C. The analysis in Appendix C is novel and shows that sampling from a binomial distribution with pseudo rewards is sufficiently optimistic.

5.3. Discussion

The regret of `Giro` is bounded from above in Theorem 2. Our bound is $O(K\Delta^{-1}\log n)$, where K is the number of arms, $\Delta = \min_{i>1}\Delta_i$ is the minimum gap, and n is the number of rounds. The bound matches the regret bound of `UCB1` in all quantities of interest.

We would like to discuss constants a and c in Theorem 2. The bound increases with the number of pseudo rewards a. This is expected, because a positive and negative pseudo rewards yield $2a + 1$ times smaller gaps than in the original problem (Section 4.2). The benefit is that exploration becomes easy. Although the gaps are smaller, `Giro` outperforms `UCB1` in all experiments in Section 6.1, even for $a = 1$. The experiments also show that the regret of `Giro` increases with a, as suggested by our bound.

The constant c in Theorem 2 is defined for all $a > 1/\sqrt{2}$ and can be large due to the term $f(b) = \exp[8b/(2-b)]$. For instance, for $a = 1$, $b = 3/2$ and $f(b) \approx e^{24}$. Fortunately, $f(b)$ decreases quickly as a increases. For $a = 2$, $b = 5/6$ and $f(b) \approx e^{5.7}$; and for $a = 3$, $b = 7/12$ and $f(b) \approx e^{3.3}$. Because `Giro` performs well empirically, as shown in Section 6.1, the theoretically-suggested value of c is likely to be loose.

6. Experiments

We conduct two experiments. In Section 6.1, we evaluate `Giro` on multi-armed bandit problems. In Section 6.2, we evaluate `Giro` in the contextual bandit setting.

6.1. Multi-Armed Bandit

We run `Giro` with three different values of a: 1, $1/3$, and $1/10$. Our regret analysis in Section 5.3 justifies the value of $a = 1$. The informal argument in Section 4.2 suggests a less conservative value of $a = 1/3$. We implement `Giro` with real $a > 0$ as follows. For each arm i, we have two histories after s pulls, with $\lfloor as \rfloor$ and $\lceil as \rceil$ pseudo rewards of each kind. The value of $\hat{\mu}_{i,t}$ is estimated from the $\lceil as \rceil$ and $\lfloor as \rfloor$ histories with probabilities $as - \lfloor as \rfloor$ and $\lceil as \rceil - as$, respectively. This interpolation seems natural.

`Giro` is compared to `UCB1` (Auer et al., 2002), Bernoulli `TS` (Agrawal and Goyal, 2013a), and `KL-UCB` (Garivier and Cappe, 2011). The prior in Bernoulli `TS` is $\mathrm{Beta}(1, 1)$. We implement `TS` and `KL-UCB` with $[0, 1]$ rewards as described in Agrawal and Goyal (2013a). Specifically, for any $Y_{i,t} \in [0, 1]$, we draw pseudo reward $\hat{Y}_{i,t} \sim \mathrm{Ber}(Y_{i,t})$ and use it instead of $Y_{i,t}$.

We experiment with two classes of K-armed bandit problems where the reward distribution P_i of arm i is parameterized by its expected reward $\mu_i \in [0, 1]$. The first class is a Bernoulli bandit, where $P_i = \mathrm{Ber}(\mu_i)$. Both `KL-UCB` and `TS` are near optimal in this class. The second class is a beta bandit, where $P_i = \mathrm{Beta}(v\mu_i, v(1-\mu_i))$ for $v \geq 1$. Both `KL-UCB` and `TS` can solve such problems, but are not statistically optimal anymore. We experiment with $v = 4$, which leads to rewards of higher variances; and $v = 16$, which leads to rewards of lower variances. The number of arms is $K = 10$ and their means are chosen uniformly at random from $[0.25, 0.75]$. The horizon is $n = 10$k rounds.

Our results are reported in Figure 1. The strong empirical performance of `Giro` is apparent. `Giro` outperforms `UCB1` in all problems and for all values of a. It also outperforms `KL-UCB` in the beta bandit for all values of a. Finally, `Giro` outperforms `TS` in the beta bandit for $a = 1/10$. Although this setting of `Giro` is purely heuristic, it shows the potential of our proposed method.

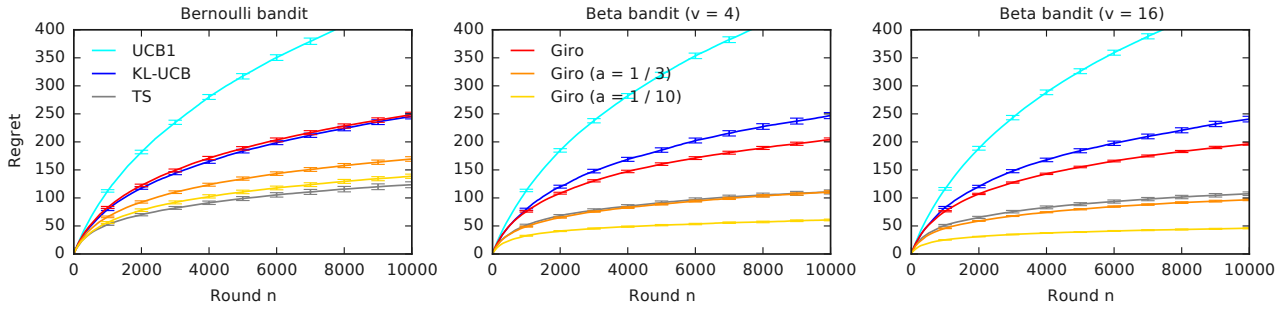

Figure 1. Comparison of `Giro` to `UCB1`, `KL-UCB`, and `TS` on three multi-armed bandit problems in Section 6.1. The regret is reported as a function of round n. The results are averaged over 100 runs. To reduce clutter, the legend is split between the first two plots.

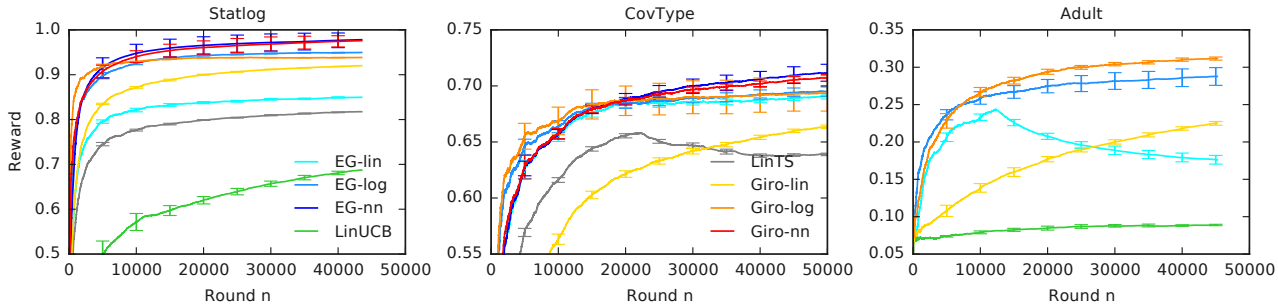

Figure 2. Comparison of `Giro` to `LinUCB`, `LinTS`, and the ϵ-greedy policy on three contextual problems in Section 6.2. The reward is reported as a function of round n. The results are averaged over 5 runs. To reduce clutter, the legend is split between the first two plots.

The goal of our experiments was to show that `Giro` performs well in general, not that it outperforms near-optimal algorithms for well-established classes of bandit problems. `Giro` is meant to be general, as shown in Section 6.2.

6.2. Contextual Bandit

We conduct contextual bandit experiments on multi-class classification problems (Agarwal et al., 2014; Elmachtoub et al., 2017; Riquelme et al., 2018), where arm $i \in [K]$ corresponds to class i. In round t, the algorithm observes context $x_t \in \mathbb{R}^d$ and pulls an arm. It receives a reward of one if the pulled arm is the correct class, and zero otherwise. Each arm maintains independent statistics that map context x_t to a binary reward. We use three datasets from Riquelme et al. (2018): Adult ($d = 94, K = 14$), Statlog ($d = 9, K = 7$), and CovType ($d = 54, K = 7$).

The horizon is $n = 50$k rounds and we average our results over 5 runs. `Giro` is compared to `LinUCB` (Abbasi-Yadkori et al., 2011), `LinTS` (Agrawal and Goyal, 2013b), and the ϵ-greedy policy (`EG`) (Auer et al., 2002). We also implemented `UCB-GLM` (Li et al., 2017). `UCB-GLM` over-explored and performed worse than our other baselines. Therefore, we do not report its results in this paper.

We experiment with three reward generalization models in `Giro` and `EG`: linear (suffix *lin* in plots), logistic (suffix *log* in plots), and a single hidden-layer fully-connected neural network (suffix *nn* in plots) with ten hidden neurons. We experimented with different exploration schedules for `EG`. The best schedule across all datasets was $\epsilon_t = b/t$, where b is set to attain 1% exploration in n rounds. Note that this tuning gives `EG` an unfair advantage over other algorithms. We tune `EG` because it performs poorly without tuning.

In `Giro`, $a = 1$ in all experiments. The parameters of the reward generalization model (line 8 in Algorithm 3) are fit in each round using maximum likelihood estimation. We solve the problem using stochastic optimization, which is initialized by the solution in the previous round. In linear and logistic models, we optimize until the error drops below 10^{-3}. In neural networks, we make one pass over the whole history. To ensure reproducibility of our results, we use public optimization libraries. For linear and logistic models, we use scikit-learn (Pedregosa et al., 2011) with stochastic optimization and its default settings. For neural networks, we use Keras (Chollet et al., 2015) with a ReLU hidden layer and a sigmoid output layer, along with SGD and its default settings. In comparison, Elmachtoub et al. (2017) and Tang et al. (2015) approximate bootstrapping by an ensemble of models. In general, our approach yields similar results to Elmachtoub et al. (2017) at a lower computational cost, and better results than Tang et al. (2015) without any tuning.

Since we compare different bandit algorithms and reward generalization models, we use the expected per-round reward in n rounds, $\mathbb{E}\left[\sum_{t=1}^{n} Y_{I_t,t}\right]/n$, as our metric. We report it for all algorithms in all datasets in Figure 2. We observe the following trends. First, both linear methods, `LinTS` and `LinUCB`, perform the worst.[1] Second, linear `Giro` and `EG` are comparable in the Statlog and CovType datasets. In the Adult dataset, `EG` does not explore enough for the relatively larger number of arms. In contrast, `Giro` explores enough and performs well. Third, the non-linear variants of `EG` and `Giro` generally outperform their linear counterparts. The most expressive model, the neural network, outperforms the logistic model in both the Statlog and CovType datasets. In the Adult dataset, the neural network performs the worst and we do not plot it. To investigate this further, we trained a neural network offline for each arm with all available data. Even then, the neural network performed worse than a linear model. We conclude that the poor performance of neural networks is caused by poor generalization and not the lack of exploration.

7. Related Work

Osband and Van Roy (2015) proposed a bandit algorithm, which they call `BootstrapThompson`; that pulls the arm with the highest bootstrap mean, which is estimated from a history with pseudo rewards. They also showed in a Bernoulli bandit that `BootstrapThompson` is equivalent to Thompson sampling. Vaswani et al. (2018) generalized this result to categorical and Gaussian rewards. In relation to these works, we make the following contributions. First, `Giro` is an instance of `BootstrapThompson` with non-parametric bootstrapping. The novelty in `Giro` is in the design of pseudo rewards, which are added after each pull of the arm and have extreme values. Second, our regret analysis of `Giro` is the first proof that justifies the use of non-parametric bootstrapping, the most common form of bootstrapping, for exploration. Finally, Algorithm 1 is more general than `BootstrapThompson`. Its analysis in Section 5.1 shows that randomization alone, not necessarily by posterior sampling, induces exploration.

Eckles and Kaptein (2014) approximated the posterior of each arm in a multi-armed bandit by multiple bootstrap samples of its history. In round t, the agent chooses randomly one sample per arm and then pulls the arm with the highest mean reward in that sample. The observed reward is added with probability 0.5 to all samples of the history. A similar method was proposed in contextual bandits by Tang et al. (2015). The key difference is that the observed reward is added to all samples of the history with random Poisson weights that control its importance. Elmachtoub et al. (2017) proposed bootstrapping with decision trees in contextual bandits. Tang et al. (2015) and Elmachtoub et al. (2017) also provided limited theoretical justification for bootstrapping as a form of posterior sampling. But this justification is not strong enough to derive regret bounds. We prove regret bounds and do not view bootstrapping as an approximation to posterior sampling.

Baransi et al. (2014) proposed a sampling technique that equalizes the histories of arms in a 2-armed bandit. Let n_1 and n_2 be the number of rewards of arms 1 and 2, respectively. Let $n_1 < n_2$. Then the value of arm 1 is estimated by its empirical mean and the value of arm 2 is estimated by the empirical mean in the bootstrap sample of its history of size n_1. Baransi et al. (2014) bounded the regret of their algorithm and Osband and Van Roy (2015) showed empirically that its regret can be linear.

8. Conclusions

We propose `Giro`, a novel bandit algorithm that pulls the arm with the highest mean reward in a non-parametric bootstrap sample of its history with pseudo rewards. The pseudo rewards are designed such that the bootstrap mean is optimistic with a high probability. We analyze `Giro` and bound its n-round regret. This is the first formal proof that justifies the use of non-parametric bootstrapping, the most common form of bootstrapping, for exploration. `Giro` can be easily applied to structured problems, and we evaluate it on both synthetic and real-world problems.

Our upper bound on the regret of randomized exploration in Theorem 1 is very general, and says that any algorithm that controls the tails of the sampling distribution p in Algorithm 1 has low regret. This shows that any appropriate randomization, not necessarily posterior sampling, can be used for exploration. In future work, we plan to investigate other randomized algorithms that easily generalize to complex problems, such as pull the arm with the highest empirical mean with randomized pseudo rewards (Kveton et al., 2019a). We believe that the key ideas in the proof of Theorem 1 can be generalized to structured problems, such as linear bandits (Kveton et al., 2019b).

History resampling is computationally expensive, because the history of the arm has to be resampled in each round. This can be avoided in some cases. In a Bernoulli bandit, `Giro` can be implemented efficiently, because the value of the arm can be drawn from a binomial distribution, as discussed in Section 4.2. A natural computationally-efficient substitute for resampling is an ensemble of fixed perturbations of the history, as in ensemble sampling (Lu and Van Roy, 2017). We plan to investigate it in future work.

[1] `LinUCB` and `LinTS` are the worst performing methods in the CovType and Adult datasets, respectively. Since we want to show most rewarding parts of the plots, we do not plot these results.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1638–1646, 2014.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013a.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 127–135, 2013b.

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

Akram Baransi, Odalric-Ambrym Maillard, and Shie Mannor. Sub-sampling for multi-armed bandits. In *Proceeding of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2014.

Francois Chollet et al. Keras. https://keras.io, 2015.

Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

Joseph Doob. *Stochastic Processes*. John Wiley & Sons, 1953.

Dean Eckles and Maurits Kaptein. Thompson sampling with the online bootstrap. *CoRR*, abs/1410.4009, 2014. URL http://arxiv.org/abs/1410.4009.

Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1): 54–75, 1986.

Adam Elmachtoub, Ryan McNellis, Sechan Oh, and Marek Petrik. A practical method for solving contextual bandit problems using decision trees. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

Sarah Filippi, Olivier Cappe, Aurelien Garivier, and Csaba Szepesvari. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23*, pages 586–594, 2010.

Aurelien Garivier and Olivier Cappe. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceeding of the 24th Annual Conference on Learning Theory*, pages 359–376, 2011.

Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *Proceedings of the 31st International Conference on Machine Learning*, pages 100–108, 2014.

Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems 30*, pages 98–108, 2017.

Jaya Kawale, Hung Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient Thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems 28*, pages 1297–1305, 2015.

Branislav Kveton, Csaba Szepesvari, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic multi-armed bandits. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019a.

Branislav Kveton, Csaba Szepesvari, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic linear bandits. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019b.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.

Lihong Li, Wei Chu, John Langford, and Robert Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2071–2080, 2017.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In *Advances in Neural Information Processing Systems 30*, pages 3258–3266, 2017.

Ian Osband and Benjamin Van Roy. Bootstrapped Thompson sampling and deep exploration. *CoRR*, abs/1507.00300, 2015. URL http://arxiv.org/abs/1507.00300.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Paat Rusmevichientong and John Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, and Tao Li. Personalized recommendation via parameter-free contextual bandits. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 323–332, 2015.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Sharan Vaswani, Branislav Kveton, Zheng Wen, Anup Rao, Mark Schmidt, and Yasin Abbasi-Yadkori. New insights into bootstrapping for bandits. *CoRR*, abs/1805.09793, 2018. URL http://arxiv.org/abs/1805.09793.

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 392–401, 2016.