Racing Thompson: an Efficient Algorithm for Thompson Sampling with Non-conjugate Priors

Yichi Zhou [1] **Jun Zhu** [1] **Jingwe Zhuo** [1]

Abstract

Thompson sampling has impressive empirical performance for many multi-armed bandit problems. But current algorithms for Thompson sampling only work for the case of conjugate priors since they require to perform online Bayesian posterior inference, which is a difficult task when the prior is not conjugate. In this paper, we propose a novel algorithm for Thompson sampling which only requires to draw samples from a tractable proposal distribution. So our algorithm is efficient even when the prior is non-conjugate. To do this, we reformulate Thompson sampling as an optimization proplem via the Gumbel-Max trick. After that we construct a set of random variables and our goal is to identify the one with highest mean which is an instance of best arm identification problems. Finally, we solve it with techniques in best arm identification. Experiments show that our algorithm works well in practice.

1. Introduction

In multi-armed bandit (MAB) problems (Lai & Robbins, 1985), an agent chooses an action (also called an arm in the literature of MAB) from an action set repeatedly, and the environment returns a reward as a response to the chosen action. The agent's goal is to maximize the cumulative reward over a period of time. In MAB, a reward distribution is associated with each arm to characterize the uncertainty of the reward. One key issue for MAB and many on-line learning problems (Bubeck et al., 2012) is to well-balance the exploitation-exploration tradeoff, that is, the tradeoff between choosing the action that has already yielded greatest rewards and the action that is relatively unexplored.

[1]Dept. of Comp. Sci. & Tech., BNRist Center, State Key Lab for Intell. Tech. & Sys., THBI Lab, Tsinghua University, Beijing, 100084, China. Correspondence to: Jun Zhu <dcszj@mail.tsinghua.edu.cn>, Yichi Zhou <vofhqn@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

As one of the most important problems in learning and decision-making in unknown environments, MAB has been studied in various settings since the seminal work (Lai & Robbins, 1985) (see (Bubeck et al., 2012) for a nice review). In this paper, we consider Bayesian bandits (Scott, 2010), which is a well-studied variant of MAB. In a Bayesian bandit, the agent has a prior distribution on the mean of the reward distribution for each action. The agent makes decisions adaptively according to the prior distributions and the past observations of each action. The most popular algorithm for Bayesian bandits is known as Thompson sampling (TS), which has a long history tracing back to (Thompson, 1933). TS has proven to be powerful in practice (Chapelle & Li, 2011) with theoretical guarantees (Agrawal & Goyal, 2012; 2017; Kaufmann et al., 2012).

Thompson sampling selects each arm randomly according to its probability to be optimal given the previous observations. Existing implementations of TS require to infer the posterior distribution (See Section 3 for details), which can be computationally intractable in sophisticated models; thereby limiting the scope of Bayesian bandits to use simple conjugate priors. However, a non-conjugate prior is very important in MAB. On one hand, non-conjugacy naturally arises because of using either a flexible prior or a flexible observation model (i.e., likelihood) to characterize the complex properties often appearing in real-world applications. For example, in the Web Advertising task (Gopalan et al., 2014), it is natural to use a Bernoulli distribution to model the binary event whether a user clicks or not, and the prior distributions of different advertisements are not independent because of the similarity between two advertisements or other factors. In this case, it is likely to have a conjugate prior that well incorporates such dependence. Another example is in (Kawale et al., 2015), where TS is used to do matrix-factorization recommendation in an online manner and their observation model is a product of two Gaussian random variables with zero mean. An inverse gamma distribution is used as the prior on their variances, which is non-conjugate with the observation model. On the other hand, if we impose an improper prior for convenience, it may lead to worse performance. For example, in the well-known stochastic Bernoulli bandits problems (Lai & Robbins, 1985), we have no prior knowledge on the

mean of these arms. In this case, Kaufmann et al. (2012) show that TS with a uniform prior achieves asymptotically optimal performance that matches the lower bound proved by Lai & Robbins (1985), while the best known theoretical result by Agrawal & Goyal (2012) with a conjugate Beta prior can lead to worse performance. Moreover, as we shall see in Section 5, an incorrect prior indeed yields worse performance in practice.

As inferring exact posteriors is often intractable in the non-conjugate case, approximate posterior inference algorithms are valuable. However, Thompson sampling requires posterior inference in an online setting which makes most popular algorithms (for example, MCMC (Propp & Wilson, 1996)) inefficient. Under the above considerations, efforts (Gopalan et al., 2014; Kawale et al., 2015; Ikonomovska et al., 2015) have been made on approximate posterior inference for TS using sequential Monte Carlo (SMC) (Del Moral & Doucet, 2014) which is relatively efficient under online setting. SMC maintains a set of particles and resamples sequentially according to the observations. However, to the best of our knowledge, there is no standard method to select the number of particles. Maintaining a large number of particles will cause inefficient computation, while a small number can be inaccurate. Overall, the above problems restrict the applications of TS significantly.

Contributions: In this paper, we present a novel racing algorithm to implement TS, which can apply to the general cases with non-conjugate priors. The algorithm approximates Thompson sampling, and only requires to sample from tractable distributions while avoiding posterior inference. Thus, the proposed algorithm is efficient. Though we also need to draw a set of samples similar to SMC, our algorithm has a simple yet powerful guideline to determine the number of samples with theoretical guarantees. And our algorithm works well in experiments.

Technically, our method is built on a novel reformulation of TS as an optimization problem by exploring the Gumbel-Max trick (Papandreou & Yuille, 2011). The goal of the optimization problem is to find the variable with the maximum expectation among a set of variables. Such a problem reduces to a best arm identification (BAI) problem (Kaufmann et al., 2015; Maron & Moore, 1997; Kalyanakrishnan et al., 2012; Jamieson & Nowak, 2014), which is a well-studied variant of stochastic multi-armed bandits. To compute the expectations involved in the BAI problem, we can freely construct a tractable prior for easy and efficient sampling, therefore avoiding the posterior inference with a non-conjugate prior.

The rest of the paper is structured as follows. We discuss related work in Section 2. Section 3 reviews some preliminary knowledge of TS. Then, we present our algorithm with the new reformulation in Section 4. Empirical studies are presented in Section 5. Finally, we conclude in Section 6.

2. Related work

Our work relates to Bayesian bandits, the Gumbel-Max trick and the best arm identification.

Bayesian bandits: Bayesian bandits have a long history dating back to (Thompson, 1933) when TS was introduced. TS is a kind of so-called probability-matching algorithms for exploitation-exploration problems. Such algorithms are relatively less known. Recently, Chapelle & Li (2011) evaluate the performance of TS compared with other famous UCB-like algorithms (Carpentier & Munos, 2011), and show that TS has a state-of-the-art performance in various tasks. Since then, many works on analyzing TS have appeared (Kaufmann et al., 2012; Agrawal & Goyal, 2012; Guha & Munagala, 2014). It turned out that TS has nice properties in a theoretical point of view. However, TS is still not very popular in practice. A possible reason is that inferring the posterior is usually intractable. This paper tries to resolve the problem.

Gumbel-Max trick: Gumbel-Max trick is a tool to connect sampling problems and optimization problems, and has been used in various problems (Chen & Ghahramani, 2016; Maddison et al., 2014; Papandreou & Yuille, 2011). The most related work is (Chen & Ghahramani, 2016), which also exploits the Gumber-Max trick for sampling a discrete random variable and relates to multi-armed bandits. But they consider the distributions with $P(x) \propto p_0(x) \prod_{i=1}^{n} p_i(x)$ which is significantly different from our problem.

Best arm identification in fixed confidence setting: This setting comes from a fundamental question about exploration and exploitation tradeoff: when can an agent stop learning and start exploiting the learned knowledge? Many algorithms have been proposed. They mainly fall into two categories: The first one is LUCB (Kalyanakrishnan et al., 2012), in which the agent pulls arms according to the confidence bound; the second is the racing algorithms which were first introduced by (Maron & Moore, 1997). In a racing algorithm, the agent maintains a set of active arms, and it pulls all active arms in each round and eliminate the suboptimal arms according to certain elimination rules.

3. Preliminaries of Thompson Sampling

Let K denote the number of arms. Each arm i is associated with a reward distribution, whose mean value is denoted by $\boldsymbol{\mu}_i$ (i.e., mean reward). We consider Bayesian bandits, which treat $\boldsymbol{\mu}_i$ as a random variable. Let π denote the prior distribution over $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$. Suppose upto time step t, the agent has chosen action i for $\tau_{i,t}$ times, and received the corresponding rewards $X_i(t) =$

$\{x_{i,1}, \cdots, x_{i,\tau_{i,t}}\}$. Let $X(t) = \{X_1(t), \cdots, X_K(t)\}$ be all the observations until time t.

Thompson sampling (TS) selects each arm randomly according to its (posterior) probability to be optimal, which is

$$\forall i \in [K], \quad P_i(t) := P\left(\boldsymbol{\mu}_i = \max_j \mu_j | X(t)\right), \qquad (1)$$

and $[K] := \{1, \cdots, K\}$ denotes the set of integers from 1 to K. Previous implementation (Chapelle & Li, 2011) of TS is outlined in Alg. 1, in which the Lines 6 and 7 essentially draw samples from $P_i(t)$. As we can see in Line 6 of Alg. 1, this implementation requires to infer the posterior distribution of the mean rewards, which is efficient if the prior is conjugate. However, in practice non-conjugate priors are more flexible in many situations such as where the arms are not independent, as discussed in the introduction.

In non-conjugate cases, directly inferring the posterior distribution is typically difficult. One possible solution is to approximate the intractable posterior with a sequential Monte Carlo (SMC) sampler, which can be done as follows: At each time step t, we maintain a set of weighted particles $\{(\xi_t^i, w_t^i)\}_{i=1}^N$, where w_t^i is the weight of particle x_t^i. Initially, these particles are sampled from the prior distribution, i.e., $\xi_1^i \sim \pi$, and the weights are equal (e.g., the unit 1). When we observed X_t, we reweight w_{t+1}^i according to the likelihood function, that is $w_{t+1}^i = w_t^i P(X_t|\xi_t^i)$. We use $P(x = \xi_t^i) = \frac{w_t^i}{\sum_{i'} w_t^{i'}}$ to approximate the posterior. Though straightforward, SMC has some shortcomings. For example, there is no standard way to choose the number of particles and when the number of observations grows up, most particles' relative weights are approaching zero (Doucet et al., 2001), it makes SMC being an inefficient approximation of the posterior. Our empirical results in Section 5 further demonstrate that SMC is not sufficient; thereby calling for a new algorithm.

4. Algorithm

We now present our algorithm. It is not easy to deal with the discrete distribution $P_i(t)$ in Eq. (1) directly, especially when the prior is non-conjugate. One key step to derive our algorithm is that we can reformulate TS as an optimization problem via the Gumbel-Max trick (Papandreou & Yuille, 2011), as detailed in Section 4.1 and followed by the racing algorithm in Section 4.2.

4.1. Thompson sampling as a bandit problem

Consider a general K-dimensional discrete distribution $P = \{P_1, \cdots, P_K\}$. Instead of directly drawing samples from P, the **Gumbel-Max** trick provides an alternative way, with which we first draw K i.i.d samples

Algorithm 1 Thompson sampling

1: Input: Prior distribution π.
2: $t = 0$.
3: Maintain sets: $X_i = \emptyset, \forall i \in [K], X = \{X_1, \cdots, X_K\}$.
4: **while** $t < T$ **do**
5: $\quad t = t + 1$.
6: $\quad$ Draw samples $\boldsymbol{\mu} \sim P(\boldsymbol{\mu}|X)$.
7: $\quad I_t = \arg\max_i \boldsymbol{\mu}_i$.
8: $\quad$ Take action I_t, and receive reward x_t, $X_{I_t} = X_{I_t} \cup \{x_t\}$.
9: **end while**

$\{\epsilon_1, \cdots, \epsilon_K\}$ from the Gumbel$(0, 1)$[1] distribution, and then set $I = \arg\max_{i \in [K]} \epsilon_i + \log P_i$. It was shown that we have the samples from the target distribution, that is, $P(I = i) = P_i$ (Kuzmin & Warmuth, 2005). So, the Gumbel-Max trick provides a nice way to turn a sampling problem to an optimization problem. It has been used in various settings (Chen & Ghahramani, 2016; Maddison et al., 2014; Papandreou & Yuille, 2011).

Applying the Gumbel-Max trick to our problem in Eq. (1), we can represent TS as the following optimization problem:

$$\begin{aligned} I_t &= \arg\max_{i \in [K]} \epsilon_i + \log P_i(t) \\ &= \arg\max_{i \in [K]} e^{\epsilon_i} P(\boldsymbol{\mu}_i = \arg\max_j \boldsymbol{\mu}_j | X(t)) \end{aligned}$$

where $\{\epsilon_i\}_{i=1}^K$ denote the set of samples from Gumbel$(0, 1)$. However, it is still hard to directly solve it.

Our key idea to solve this problem efficiently is to construct a tractable distribution and further turn this problem as a best arm identification (BAI) problem. Specifically, by introducing an arbitrary proposal distribution, B_t, over the state space of $\boldsymbol{\mu}$ (we will introduce how to choose suitable proposal for our problem in Section 4.1.1.), we can reformulate the problem as follows:

$$\begin{aligned} I_t &= \arg\max_{i \in [K]} e^{\epsilon_i} \int_{\boldsymbol{\mu}} P(\boldsymbol{\mu}|X(t)) \mathbb{1}[\boldsymbol{\mu}_i = \max_j \boldsymbol{\mu}_j] \\ &= \arg\max_{i \in [K]} e^{\epsilon_i} \int_{\boldsymbol{\mu}} B_t(\boldsymbol{\mu}) \mathbb{1}[\boldsymbol{\mu}_i = \max_j \boldsymbol{\mu}_j] \frac{P(\boldsymbol{\mu}|X(t))}{B_t(\boldsymbol{\mu})} \\ &= \arg\max_{i \in [K]} \mathbb{E}_{\boldsymbol{\mu} \sim B_t} \left[e^{\epsilon_i} \mathbb{1}[\boldsymbol{\mu}_i = \max_j \boldsymbol{\mu}_j] \frac{P(\boldsymbol{\mu}|X(t))}{B_t(\boldsymbol{\mu})} \right] \\ &= \arg\max_{i \in [K]} \mathbb{E}_{\boldsymbol{\mu} \sim B_t} \left[e^{\epsilon_i} \mathbb{1}[\boldsymbol{\mu}_i = \max_j \boldsymbol{\mu}_j] \frac{P(X(t)|\boldsymbol{\mu})}{B_t(\boldsymbol{\mu})} \right] \end{aligned} \qquad (2)$$

[1] If $\epsilon_i \sim$ Gumbel$(0, 1)$, then $P(\epsilon = x) \propto e^{-(x + e^{-x})}$. Moreover, it is easy to sample from Gumbel$(0, 1)$—simply draw u from the uniform distribution $U[0, 1]$ and set ϵ as $-\log(-\log u)$.

From the Gumbel-Max theory, we know that I_t follows our target posterior distribution, that is, $I_t \sim P(t)$ (Papandreou & Yuille, 2011). Essentially, the problem in Eq. (2) is to find the variable with maximum expectation and we can use Monte-Carlo methods to estimate the expectation efficiently. As in our case, each variable corresponds to an arm, and this is known as a best arm identification (BAI) problem (Jamieson & Nowak, 2014). The benefit of our formulation is that we only need to draw samples from proposal B_t, which can be done efficiently, in order to estimate the expectations.

We solve the above BAI problem in the popular fixed-confidence setting (Jamieson & Nowak, 2014): for $\delta \in (0, 1)$ and $\sigma > 0$, an algorithm is called (δ, σ)-PAC (Kaufmann et al., 2015) if and only if with probability at least $1 - \delta$, it identifies arm i such that $\mu_i > \max_{j\in[K]} \mu_j - \sigma$. This setting provides a simple and practical method to determine the number of samples we need to draw from the proposal B_t: we can stop the sampling process if we are sure enough that we have identified a sufficiently good arm. Following lemma shows a (δ, σ)-PAC algorithm is asymptotically good.

Lemma 1. *Let $P_i(\delta, \sigma, t)$ be the sampling distribution of an (δ, σ)-PAC algorithm for Eq. (2), then the total variation between $P_i(\delta, \sigma, t)$ and $P_i(t)$ converges to 0 asymptotically, that is:*

$$\lim_{\delta,\sigma\to 0} \sum_i |P_i(\delta, \sigma, t) - P_i(t)| = 0$$

Proof. According to the definition, a (δ, σ)-PAC algorithm outputs i such that $e^{\epsilon_i} P_i > e^{\epsilon_j} P_j - \sigma, \forall j$ with probability at least $1 - \delta$. Let event $\mathcal{E}$ denote the algorithm outputs i : $e^{\epsilon_i} P_i > e^{\epsilon_j} P_j - \sigma, \forall j$ successfully. We know that $P(\mathcal{E}) > 1 - \delta$. Let $\mathcal{A} \in [K]$ be the output of this algorithm, $\mathcal{G} \in [K]$ be an arbitrary arm such that $e^{\epsilon_\mathcal{G}} P_\mathcal{G} > e^{\epsilon_j} P_j - \sigma, \forall j \in [K]$. We use following process to generate a random variable $\mathcal{K}$: If $\mathcal{E}$ happens, $\mathcal{K} = \mathcal{A}$, else $\mathcal{K} = \mathcal{G}$. We have:

$$\begin{aligned}
&\sum_i |P_i(\delta, \sigma, t) - P_i(t)| \\
&\leq \sum_i |P_i(\delta, \sigma, t) - P(\mathcal{K} = i)| + |P_i(t) - P(\mathcal{K} = i)| \\
&\leq \delta + \sum_i |P_i(t) - P(\mathcal{K} = i)|
\end{aligned}$$

Now we only need to bound the second term. $\sum_i |P_i(t) - P(\mathcal{K} = i)|$ is independent with δ and by definition, $e^{\epsilon_\mathcal{K}} P_\mathcal{K} > e^{\epsilon_j} P_j - \sigma$ is always true. So we have:

$$P(\mathcal{K} = i) \leq Pr[e^{\epsilon_i} P_i > \max_j e^{\epsilon_j} P_j - \sigma]$$

and

$$P(\mathcal{K} = i) \geq Pr[e^{\epsilon_i} P_i > \max_j e^{\epsilon_j} P_j + \sigma]$$

Recall that $\epsilon_1, \cdots, \epsilon_K$ are i.i.d Gumbel random variables. According to the definition of Gumbel distribution and with algebras, we have:

$$\begin{aligned}
&Pr[\exp\{\epsilon_i\} P_i > \max_j \exp\{\epsilon_j\} P_j - \sigma] \\
&= \int_{\epsilon_i=-\infty}^{+\infty} Pr(\epsilon_i) \prod_{j\neq i} Pr[\epsilon_j < \log \frac{e^{\epsilon_i} P_i + \sigma}{P_j}] \\
&= \int_{\epsilon_i=-\infty}^{\infty} \exp(-\epsilon_i - \exp(-\epsilon_i)) \exp\left(-\frac{1 - P_i}{e^{\epsilon_i} P_i + \sigma}\right)
\end{aligned}$$

Above function is continuous and monotonically increasing with respect to σ, so as $\sigma \to 0$, we have:

$$\begin{aligned}
&\lim_{\sigma\to 0} Pr[\exp\{\epsilon_i\} P_i > \max_j \exp\{\epsilon_j\} P_j - \sigma] \\
&= \int_{\epsilon_i=-\infty}^{\infty} \exp\left(-\epsilon - \frac{1}{P_i} \exp(-\epsilon)\right) = P_i
\end{aligned}$$

Similarly, we have $\lim_{\sigma\to 0} P(\mathcal{K} = i) \geq P_i(t)$. Now we complete the proof. □

4.1.1. Choose proposal

Now we consider how to choose a suitable proposal B_t. The first choice is to let $B_t = \pi$ directly as in SMC. However, as the number of observations grows, the distributions $\pi(\boldsymbol{\mu})$ and $P(\boldsymbol{\mu}|X(t))$ will become ill-matched in most cases (Doucet et al., 2001). This will violate the assumption of our stopping rule (see Section 4.2 for details).

A better proposal can be constructed by the prior-swapping trick (Neiswanger & Xing, 2017). The key observation of the prior swapping trick is that for many practical distributions, there exists a convenient prior that allows for efficient posterior inference. For example, suppose $P(x|\boldsymbol{\mu}_i)$ is a Bernoulli distribution, then $Beta$ distribution is the conjugate prior. Suppose $\pi' = \{\pi'_1, \cdots, \pi'_K\}$ is such a prior that allows for efficient posterior inference, we use the corresponding posterior $\pi'(\boldsymbol{\mu})P(X(t)|\boldsymbol{\mu})$ as our proposal B_t. When the number of observation grows, B_t still matches $P(t)$ well. The reason is that by Bayes' rule, we have $P(\boldsymbol{\mu}|X) \propto \pi(\boldsymbol{\mu}) \prod_{i=1}^n P(X_i|\boldsymbol{\mu})$ where n is the number of observations. When n grows up, likelihood $\prod_{i=1}^n P(X_i|\boldsymbol{\mu})$ "dominates" the posterior, and our proposal B_t carries all information of the likelihood term.

For many widely-used distributions, there exist such priors. For example, any exponential family distribution exists a conjugate prior. And we can efficiently draw samples from the posterior distributions for many of them, see (George et al., 1993) for more details. Beyond exponential family distributions with conjugate priors, there are tractable distributions with other priors. For example, many 1-dimensional exponential family distributions with a non-informative Jeffreys prior is tractable (Jaynes, 1968). The Jeffreys prior

Name	distribution	prior	posterior
Bernoulli	$p^x(1-p)^{1-x}$	$Beta(1,1)$	$Beta(1+s, 1+T-s)$
Exponential	$\lambda e^{-\lambda x}$	$\Gamma(1,1)$	$\Gamma(1+T, 1+s)$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\{-\frac{(x-\mu)^2}{2}\}$	$\propto 1$	$\mathcal{N}(\frac{s}{T}, \frac{1}{T})$
Poisson	$\frac{\lambda^x e^{-\lambda}}{x!}$	$\propto \frac{1}{\sqrt{\lambda}}$	$\Gamma(\frac{1}{2}+s, T)$

Table 1. Examples of distributions with Jeffreys prior. s is the sum of T observations.

is proportional to the square root of the Fisher information matrix and maybe improper (e.g., the Jeffreys prior for Gaussian is uniform in an infinite space). Some tractable and representative exponential family distributions with Jeffreys prior are listed in Table 1. So in the sequel of this paper, we use such posterior with a false prior as our proposal.

4.2. The racing algorithm

For the clarity of notations, we let

$$f_i(\boldsymbol{\mu}, t) = e^{\epsilon_i}\mathbb{1}[\boldsymbol{\mu}_i = \max_j \boldsymbol{\mu}_j]\frac{P(X(t)|\boldsymbol{\mu})}{B_t(\boldsymbol{\mu})}$$

as the function within the expectation and $B_t(\boldsymbol{\mu})$ is the proposal distribution at time t. We further use $v_i = f_i(\boldsymbol{\mu}, t)$ to denote a random variable, where $\boldsymbol{\mu} \sim B_t$. Recall that our goal is to identify the arm with the maximum expectation $I_t = \arg\max_i \mathbb{E}[v_i]$. Suppose we have a set of samples $d = \{d_1, \cdots, d_m\}$ where $d_j \sim B_t$. We use $f_{i,m} = \frac{1}{m}\sum_{j=1}^m f_i(d_j, t)$ as our unbiased estimator of $\mathbb{E}[v_i]$.

Recall that the goal of a BAI problem is to identify the one with the highest expectation among a set of random variables. Following (Kaufmann et al., 2015), a practical BAI algorithm in the fixed confidence setting typically consists of:

- Policy: given a sequence of past observations, a policy determines which arms to pull.
- Stopping rule: a stopping rule can be described as a series of observation sets $\mathcal{F}_t, t \in \mathbb{N}_+$[2] , where F_t is a set of observations. When an element $o \in \mathcal{F}_t$ is observed, the policy stops sampling.
- Recommendation rule: a recommendation rule is usually to recommend the best arm. A BAI algorithm usually recommends the arm with the highest empirical mean.

Our algorithm imitates the racing algorithms (Maron & Moore, 1997; Even-Dar et al., 2006) for BAI problems. In racing algorithms, people maintain a set of arms as the candidates of the best arm. The policy of a racing algorithm is to draw a sample from the underlying distribution of each remained arm during each round. And then eliminate the suboptimal arms if the gap between the empirical means of the suboptimal arm and the maximum one is bigger than a threshold function. A racing algorithm stops if and only if only one arm is not eliminated. The reason we choose racing algorithm is that we can compute $f_{i,m}$ with the same d, so drawing a sample from B_t is essentially pulling all arms at the same time. Our algorithm is shown in Alg. 2. In lines 7-15 of Alg. 2, we solve the BAI problem via a racing algorithm: we sample $\hat{\boldsymbol{\mu}} \sim B_t$ repeatedly until the empirically best arm i_1 is better than others i_2 significantly, e.g, larger than a threshold function defined by $\beta(m, \delta)$.

Theorem 1 guarantees that Alg. 2 is (δ, σ)-PAC.

Theorem 1. *If the threshold function $\beta(m, \delta)$ satisfies the following condition,*

$$P\left(\exists m > 0 : |f_{i,m} - \mathbb{E}[v_i]| > \beta(m, \delta)\right) < \delta. \quad (3)$$

then in Alg. 2, at each time t, with probability at least $1 - K\delta$, $\mathbb{E}[v_{I_t}] > \mathbb{E}[\bar{f}_i] - \sigma$ for all i.

Proof. We exploit standard arguments to prove the theorem. When Line 13 is executed, and I is a bad arm, that is $\mathbb{E}f_I < \mathbb{E}\bar{f}_i - \sigma$. By $\bar{f}_I - \bar{f}_{i^*} > 2\beta(m, \delta) - \sigma$. It is easy to see that at least one of the following two events happens: $\bar{f}_I - \beta(m, \delta) > \mathbb{E}f_I$ or $\bar{f}_{i^*} + \beta(m, \delta) - \sigma < \mathbb{E}f_{i^*} - \sigma$. With InEq. (3) and the union bound, we complete the proof. □

As stated in Theorem 1, the only requirement on the threshold function $\beta(m, \delta)$ is that it should satisfy the condition in InEq. (3). In fact, there are various threshold functions satisfying condition InEq. (3) under different assumptions (Kaufmann et al., 2015; Jamieson & Nowak, 2014). It is significant to choose a suitable assumption for our problem in Eq. (2) and pick a corresponding threshold function since the threshold function determines the number of particles used in Alg. 2. We adopt the threshold function for Gaussian bandits (Kaufmann et al., 2015) by considering the following two aspects: (1) the random variable in Eq. (2) can be unbounded and Gaussian bandits consider random variables with an unbounded support; (2) according to the law of large numbers, as the number of observations increases,

[2] $\mathbb{N}_+$ is the set of positive integers.

Algorithm 2 Racing Thompson

Input: Prior distribution π, parameters δ, σ, and time horizon T.
$t = 0$.
$X_i = \emptyset, \forall i \in [K], X = \{X_1, \cdots, X_K\}$.
while $t < T$ **do**
  Draw K i.i.d samples ϵ_i from the Gumbel$(0, 1)$ distribution.
  $m = 1, d = \emptyset$.
  loop
    Draw a sample $\hat{\boldsymbol{\mu}} \sim B_t$, $d = d \cup \{\hat{\boldsymbol{\mu}}\}$.
    $m = m + 1$.
    $\forall i, \bar{f}_i = \frac{1}{m} \sum_{j=1}^{m} f_i(d_j, t)$.
    Identify the best arm $i_1 = \arg\max_{i \in [K]} \bar{f}_i$ and the second best $i_2 = \arg\max_{i \in [K] \setminus \{i_1\}} \bar{f}_i$.
    if $\bar{f}_{i_1} - \bar{f}_{i_2} > 2\beta(m, \delta) - \sigma$ **then**
      Break the loop.
    end if
  end loop
  $I_t = \arg\max_i \frac{1}{m} \sum_{j=1}^{m} f_i(d_j, t)$.
  Take action I_t, and receive reward x_t, $X_{I_t} = X_{I_t} \cup \{x_t\}$.
end while

the variance of random variables in Eq. (2) converges to 0. Thus, it is natural to take variance into consideration. And Kaufmann et al. (2015) consider variance in their threshold function and design a near optimal algorithm for Gaussian bandits. We present their results in Lemma 2.

Lemma 2 ((Kaufmann et al., 2015)). *Let $x_1, x_2, \cdots$ be a series of i.i.d variables sampled from a Gaussian distribution with mean $\boldsymbol{\mu}$ and variance η. Let $\bar{\boldsymbol{\mu}}_m = \frac{1}{m} \sum_{i \in [m]} x_i$. Define*

$$\beta(m, \delta) = \sqrt{\frac{2\eta}{m}(\log \frac{1}{\delta} + 3 \log\log \frac{1}{\delta} + (3/2) \log\log(em))}. \tag{4}$$

Then the following inequality holds:

$$P(\exists m > 1, |\bar{\boldsymbol{\mu}}_m - \boldsymbol{\mu}| > \beta(m, \delta)) < \delta.$$

In practice, we do not know the variance in advance. So we empirically estimate it instead. As we can see in Section 5, our algorithm works well.

4.2.1. Independent prior π

For a typical racing algorithm (Maron & Moore, 1997), if we have confirmed that an arm is suboptimal, then we'll never pull it again. Thus, suppose at a round, there remains s active arms, then the running time of a typical racing algorithm is $O(s)$. However, it is clear that the running time of Alg. 2 is $O(K)$ per round. This is because when π is a dependent prior, the estimator $f_{i,m}$ of an active arm i relies on all arms.

Fortunately, when π is an independent prior (it is possible that π is still non-conjugate), that is $\pi = \prod_{j \in [K]} \pi_k$, we do not need to care about the eliminated arms anymore. For clarity, let $\mathcal{I}_t$ denote the set of active arms at round t. With algebras, it is clear that we only need to identify arm:

$$I_t = \arg\max_{i \in \mathcal{I}_t} \int \prod_{j \in \mathcal{I}_t} \pi_j(\boldsymbol{\mu}_j) e^{\epsilon_i} \mathbb{1}\left[\boldsymbol{\mu}_i = \arg\max_{j \in \mathcal{I}_t} \boldsymbol{\mu}_j\right] \prod_{j \in \mathcal{I}_t, s \in [\tau_j]} P_j(x_{j,s} | \boldsymbol{\mu}_j) d\boldsymbol{\mu}.$$

Obviously, I_t is independent from the arms not in $\mathcal{I}_t$. With the above ideas, the per round running time of Alg. 2 can be reduced from $O(K)$ to $O(|\mathcal{I}_t|)$.

5. Experiments

In this section, we empirically compare Alg. 2 with three implementations of Alg. 1 which are efficient in online posterior inference:

- **Thompson**: the first method is the vanilla Thompson sampling (TS) (i.e., the exact version of Alg. 1). As stated before, this method only works when using a conjugate prior. For the problems with a non-conjugate prior, we will use a conjugate surrogate, which is improper, so that we can still infer the posterior efficiently in closed-form and apply the standard TS;
- **Sequential Monte-Carlo (SMC)**: the second one is to use sequential Monte-Carlo to approximate the posterior;
- **Prior swapping (PS)** (Neiswanger & Xing, 2017): the last one is to use the prior-swapping trick to approximate the posterior. The difference between PS and SMC is that PS uses the posterior with a false but convenient prior to be proposal while SMC uses a true prior as the proposal.

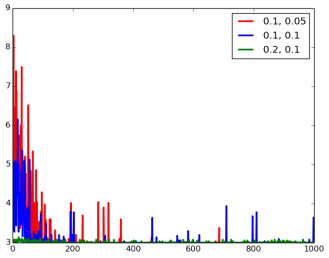
Recall that the agent selects arm I_t at time step t, and the goal is to maximize the expected cumulative reward $\mathbb{E}_{\boldsymbol{\mu} \sim \pi} \sum_{t=1}^{T} \boldsymbol{\mu}_{I_t}$, where $\boldsymbol{\mu}_i$ is the unknown mean of arm i, and $\boldsymbol{\mu} \sim \pi$. It is easy to see that maximizing the cumulative reward is equivalent to minimizing the regret:

$$\mathbb{E}_{\boldsymbol{\mu} \sim \pi}\left[T \max_{i \in [K]} \boldsymbol{\mu}_i - \sum_{t=1}^{T} \boldsymbol{\mu}_{I_t}\right]. \tag{5}$$

We follow the setting in (Chapelle & Li, 2011) and compare the regret of Alg. 2 with baselines in all our experiments.

	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$
$\delta = 0.1$	4.94	3.51	2.97
$\delta = 0.2$	4.01	3.14	2.9
$\delta = 0.3$	3.93	2.99	2.9

(a)



(b)

Figure 1. Empirical evaluation of the relationship between the number of particles and parameters δ, σ where (**a**) the averaged number of particles for each arm for various settings with the number of time stepequaling to 100 and (**b**) the number of particles for each arm varies over time.

5.1. Sensitivity of parameters

Our racing algorithm has two parameters δ, σ as shown in Alg. 2, where these parameters balance the number of samples we need to draw and the accuracy of Alg. 2 (Please see Section 4.2). We first empirically analyze the relationship between the number of particles and parameters δ, σ in Alg 2. We do experiments on Bernoulli bandits, i.e. $P(x|\boldsymbol{\mu}_i)$ is a Bernoulli distribution. We consider a Bayesian bandit problem with 5 arms. We use $Beta(5, 5)$ as the prior of all arms and set the number of time steps as $t = 1000$. We repeat the experiment for 10 times and present the average results in Fig. 1.

Fig. 1 (a) shows that the smaller the parameters δ, σ are, the more particles we use. This is consistent with the definition of δ and σ (Please see Section 4.1). Fig. 1(b) presents the number of particles varying over time. As we can see, the number of particles decreases as the number of time steps increases. This is because the variance of the random variable associated with each arm decreases as the number of observations grows up and the Alg. 2 stops early when the variance is small (See Section 4.2 for more details).

5.2. Regret compared with vanilla Thompson sampling

In order to evaluate the performance of Alg. 2 in terms of regret, we first compare Alg. 2 with Thompson sampling in the bandit problems with conjugate priors. We do experiments on Bernoulli bandits. To make Alg. 1 computationally efficient, we use the Beta distribution which is the conjugate prior so that the standard TS can apply. There are 10 arms, and the prior of the i-th arm follows $Beta(\cdot|a, b)$, where a and b are uniformly selected from the interval $(1.0, 10.0)$. For Alg. 2, we use the posterior distribution with a $Beta(1, 1)$ prior as our proposal (B_t). We repeat for 100 times and present the average results in Fig. 2(a). Fig. 2(a) shows that our racing Alg. 2 has similar performance in terms of regret compared with the vanilla Thompson sampling (i.e., Alg. 1) for parameters $(\sigma, \epsilon) = (0.05, 0.1)$.And on average, Alg. 2 uses about 3.5 samples per arm in each time step.

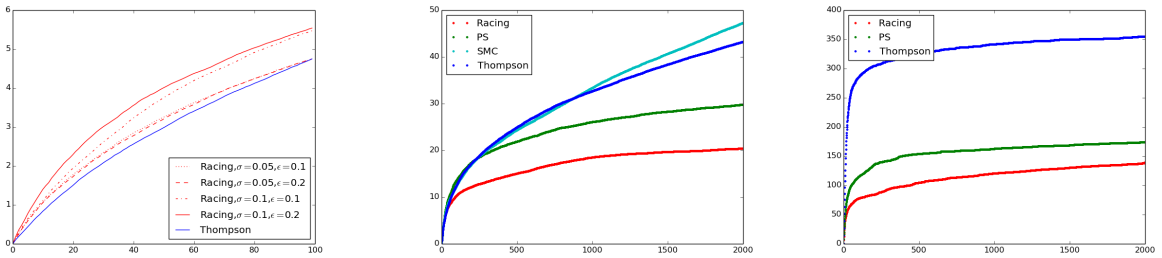
We set δ and σ to be 0.1 and 0.05 respectively in following experiments as under this setting, Alg. 2 has similar regret to vanilla Thompson sampling.

5.3. Non-conjugate prior

In this section, we evaluate Alg. 2 and the baselines on the bandits with non-conjugate priors. We consider two representative bandit problems: Bernoulli bandits and Gaussian bandits. Notice that Alg. 2, SMC and PS are all particle-based methods, that is, we use samples from proposal distributions to estimate the target distribution, i.e., $P(t)$. To prove the efficiency of Alg. 2, we use much more particles for SMC and PS in our experiments, so that their running time is longer than Alg. 2. And we'll show that even though SMC and PS have longer running time, they have worse performances compared with Alg. 2 in terms of the regret.

5.3.1. Bernoulli bandit with non-conjugate prior

We first evaluate the performance of Alg. 2 for a non-conjugate prior on Bernoulli bandits. We also do experiments on synthetic data with 10 arms. Suppose there is a real-valued vector u_i associated with each arm, and the L_2-norm of u_i is 1. These vectors can be interpreted as features. For convenience, suppose the prior is a 10-dimensional Gaussian distribution with each dimension's mean is 0.5 and we restrict the value within $[0, 1]$. We define the covariance matrix Σ of the prior distribution as follows: $\Sigma_{i,i} = 1$ for all i and $\Sigma_{i,j} = u_i \cdot u_j$, where $x \cdot y$ denotes the inner product between vectors x and y. Similar to last experiment, we set the number of time steps at 2000. As mentioned before, we run Alg. 2 with parameters $\delta = 0.1, \sigma = 0.05$, and in this experiment, Alg. 2 uses about 3.0 particles per arm. There is no principled method to determine the number of particles for PS and SMC. To compare these algorithms fairly, we use more particles for PS and SMC. More specifically, for PS and SMC, we use 10 and 20 particles for each arm respectively. For Thompson, we consider a uniform prior which allows efficient posterior inference. Results are presented in Fig. 2 (b). As shown in Fig. 2(b), Alg. 2 outperforms all baselines significantly even though PS and SMC use much more particles compared with Alg. 2.



(a) Conjugate prior (b) Non-conjugate Bernoulli bandit (c) Non-conjugate Gaussian bandit

Figure 2. The regret of bandits with (**a**) Bernoulli bandits with a conjugate prior; (**b**) Bernoulli bandits with a non-conjugate prior and (**c**) Gaussian bandits a non-conjugate prior.

5.3.2. Gaussian bandit with non-conjugate prior

Our last experiment is on Gaussian bandits with non-conjugate priors. There are 10 arms, and the mean of the i-th arm is uniformly selected from $[-10, 10]$. We set the number of time steps at 2000. We first run Alg. 2 with parameters $\delta = 0.1, \sigma = 0.05$, which uses 3.5 particles on average for each arm per round. For PS and SMC, we use 10 and 20 particles for each arm respectively. And for Thompson sampling, we use an improper prior $\mathcal{N}(0, 5)$. The results are averaged over 100 runs. We present the results except that of SMC in Fig. 2(c), since it has a very bad performance in this experiment, and the averaged regret of SMC is about 1200. We can see that Alg. 2 also outperforms Thompson and PS even though PS uses much more particles to approximate the posterior.

6. Conclusion and future work

We present an efficient racing algorithm for Thompson sampling with general non-conjugate priors. Our method is built on a new reformulation of Thompson sampling as a best arm identification problem based on the Gumbel-Max trick. Our racing algorithm has a theoretical guarantee. And we show that even if the prior is non-conjugate, we can implement Thompson sampling efficiently and not hurt the performance concurrently.

We believe our work enlarges the applicable area where Thompson sampling was hard to be applied due to the non-conjugate priors in the past. In real-world applications, non-conjugate prior appears naturally, for example, when using a dependent prior to capture the similarities between articles in article recommendation. We think it is meaningful to explore the usage of non-conjugate priors on such tasks. We hope our work encourages more efforts on applying Bayesian Bandits with non-conjugate priors to more applications.

7. Acknowledgements

This work was supported by NSFC Projects (Nos. 61620106010, 61621136008, 61332007), Beijing NSF Project (No. L172037), Tiangong Institute for Intelligent Computing, NVIDIA NVAIL Program, Siemens and Intel.

References

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, pp. 39–1, 2012.

Agrawal, S. and Goyal, N. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5): 30, 2017.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5 (1):1–122, 2012.

Carpentier, A. and Munos, R. Finite time analysis of stratified sampling for monte carlo. In *Advances in Neural Information Processing Systems*, pp. 1278–1286, 2011.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Chen, Y. and Ghahramani, Z. Scalable discrete sampling as a multi-armed bandit problem. In *International Conference on Machine Learning*, pp. 2492–2501, 2016.

Del Moral, P. and Doucet, A. Particle methods: An introduction with applications. In *ESAIM: Proceedings*, volume 44, pp. 1–46. EDP Sciences, 2014.

Doucet, A., De Freitas, N., and Gordon, N. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pp. 3–14. Springer, 2001.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

George, E. I., Makov, U., and Smith, A. Conjugate likelihood distributions. *Scandinavian Journal of Statistics*, pp. 147–156, 1993.

Gopalan, A., Mannor, S., and Mansour, Y. Thompson sampling for complex online problems. In *ICML*, volume 14, pp. 100–108, 2014.

Guha, S. and Munagala, K. Stochastic regret minimization via thompson sampling. In *COLT*, pp. 317–338, 2014.

Ikonomovska, E., Jafarpour, S., and Dasdan, A. Real-time bid prediction using thompson sampling-based expert selection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1869–1878. ACM, 2015.

Jamieson, K. and Nowak, R. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pp. 1–6. IEEE, 2014.

Jaynes, E. T. Prior probabilities. *IEEE Transactions on systems science and cybernetics*, 4(3):227–241, 1968.

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 655–662, 2012.

Kaufmann, E., Korda, N., and Munos, R. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pp. 199–213. Springer, 2012.

Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 2015.

Kawale, J., Bui, H. H., Kveton, B., Tran-Thanh, L., and Chawla, S. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, pp. 1297–1305, 2015.

Kuzmin, D. and Warmuth, M. K. Optimum follow the leader algorithm. In *International Conference on Computational Learning Theory*, pp. 684–686. Springer, 2005.

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

Maddison, C. J., Tarlow, D., and Minka, T. A* sampling. In *Advances in Neural Information Processing Systems*, pp. 3086–3094, 2014.

Maron, O. and Moore, A. W. The racing algorithm: Model selection for lazy learners. In *Lazy learning*, pp. 193–225. Springer, 1997.

Neiswanger, W. and Xing, E. Post-inference prior swapping. In *International Conference on Machine Learning*, pp. 2594–2602, 2017.

Papandreou, G. and Yuille, A. L. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 193–200. IEEE, 2011.

Propp, J. G. and Wilson, D. B. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random structures and Algorithms*, 9(1-2):223–252, 1996.

Scott, S. L. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

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