Semi-Parametric Sampling for Stochastic Bandits with Many Arms

Mingdong Ou, Nan Li, Cheng Yang, Shenghuo Zhu, Rong Jin
Alibaba Group, Hang Zhou, China
{mingdong.omd, nanli.ln, charis.yangc, shenghuo.zhu, jinrong.jr}@alibaba-inc.com

Abstract

We consider the stochastic bandit problem with a large candidate arm set. In this setting, classic multi-armed bandit algorithms, which assume independence among arms and adopt non-parametric reward model, are inefficient, due to the large number of arms. By exploiting arm correlations based on a parametric reward model with arm features, contextual bandit algorithms are more efficient, but they can also suffer from large regret in practical applications, due to the reward estimation bias from mis-specified model assumption or incomplete features. In this paper, we propose a novel Bayesian framework, called *Semi-Parametric Sampling* (SPS), for this problem, which employs semi-parametric function as the reward model. Specifically, the parametric part of SPS, which models expected reward as a parametric function of arm feature, can efficiently eliminate poor arms from candidate set. The non-parametric part of SPS, which adopts non-parametric reward model, revises the parametric estimation to avoid estimation bias, especially on the remained candidate arms. We give an implementation of SPS, *Linear SPS* (LSPS), which utilizes linear function as the parametric part. In semi-parametric environment, theoretical analysis shows that LSPS achieves better regret bound (i.e. $\tilde{O}(\sqrt{N}^{1-\alpha}d^{\alpha}\sqrt{T})$ with $\alpha \in [0, 1]$) than existing approaches. Also, experiments demonstrate the superiority of the proposed approach.

1 Introduction

In the stochastic bandit problem (Bubeck, Cesa-Bianchi, and others 2012), the agent sequentially selects arms from a finite candidate set and receives corresponding stochastic reward. The objective is to maximize the expected cumulative reward up to final time step T. As the expected reward of each arm is unknown, the agent has to face the exploration-exploitation dilemma. On one hand, the agent can exploit historical knowledge (including selected arms, stochastic rewards and state of arms) to select the arm with largest estimated expected reward. This will guarantee relatively large reward in current time step, but may suffer reward estimation error because of insufficient knowledge, then miss the real optimal arm. On the other hand, the agent can explore arms to collect knowledge for more accurate estimation and identification of optimal arm. Thus, the agent can expect to achieve larger reward in the future. However, it will probably obtain small reward in current time step. Such problem is ubiquitous in many practical applications, including online advertisement (Schwartz, Bradlow, and Fader 2017), personalized recommendation (Li et al. 2010), medical treatment (Wang et al. 2018). There are often many newly emerging items with unknown reward to explore in these applications. For example, on some news platforms, more than one hundred thousand news articles are published every day and we does not know their popularity. Since the number of new items is often large, stochastic bandits with many arms becomes important.

Existing bandits algorithms can be generally categorized into two groups: multi-armed bandit (MAB) algorithms and contextual bandit algorithms. Classic MAB algorithms, with *Upper Confidence Bound* (Auer, Cesa-Bianchi, and Fischer 2002) and *Thompson sampling* (Thompson 1933) as representatives, assume independence among candidate arms and estimate the expected reward of an arm by empirical mean of historical stochastic rewards. The reward estimation can be unbiased with sufficient data. However, as the arms are independent, the agent has to explore all the arms to collect data, leading to $\Omega(\sqrt{NT})$ regret bound, which is dependent on the number of arms N. This makes such group of algorithms inefficient for large number of candidate arms. Contextual bandit algorithms, with *LinUCB* (Chu et al. 2011) and *LinTS* (Agrawal and Goyal 2013) as representative examples, exploit the correlation among arms based on a parametric reward function of context features, and their regret is usually free of arm number, indicating that they are efficient for large arm set. In the following, contextual bandits are called *parametric* methods, since they often employ a parametric function to model the reward expectation, and MAB algorithms as *non-parametric* methods since they does not make any parametric assumption of the reward. The fundamental assumption of parametric methods is that the expected reward can be well expressed as a function of context features. However, it is generally difficult to specify a correct model and difficult to perfectly characterize the context with numeric features, this makes the assumption impractical, leading to large regret in many applications. For example, it has been shown in (Gopalan, Maillard, and Zaki 2016; Ghosh, Chowdhury, and Gopalan 2017) that the regret can be linear, i.e. $\tilde{O}(T)$, in the case of mis-specified model.

Therefore, designing efficient algorithms for stochastic bandit problem with many arms is still a open problem.

In this paper, we propose a novel framework, called *Semi-Parametric Sampling* (SPS), to solve the problem. Specifically, SPS assumes that the stochastic rewards of arms are generated by a hierarchical process. In this process, the stochastic reward of an arm is assumed to be a sample from a distribution whose mean is the expected reward variable of the arm. It assumes a parametric prior distribution to generate the parametric part of expected reward, and a non-parametric likelihood distribution to generate expected reward deviating from parametric part. As the parametric prior distribution exploits the correlation on arm feature, it will concentrate fast with the increasing of collected stochastic rewards. Thus, it can provide a good base of expected reward. Informally, the candidate set of optimal arm will be reduced rapidly. Then the exploration on reduced candidate set will cost much less to correctly identify the optimal arm. Therefore, SPS will be efficient because of rapid reduction on candidate optimal arm set, and unbiased because of non-parametric estimation on expected rewards of candidate optimal arms. We give an implementation of SPS when the parametric part of expected reward is linear function, called *Linear Semi-Parametric Sampling* (LSPS). We also prove an upper Bayesian regret bound, $\tilde{O}(\sqrt{N}^{1-\alpha}d^{\alpha}\sqrt{T})$ with $\alpha \in [0, 1]$. Compared to existing bandit algorithms, LSPS achieves $(\sqrt{N}/d)^{\alpha}$-order improvement in semi-parametric environment, where α is based on the level of parametric estimation bias. We also conduct experiments on synthetic data which shows that LSPS will achieve lower regret.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 and 4 formally formulate the problem and present the algorithm. Section 5 gives a bayesian regret bound. Section 6 summarizes the experiments, and Section 7 concludes this work with future directions.

2 Related Work

As mentioned above, most of the existing bandit algorithms can be partitioned into two classes: classic MAB algorithms and contextual bandit algorithms. Classic MAB algorithms was first proposed in the literature. They work with strategies, such as UCB (Agrawal 1995; Auer, Cesa-Bianchi, and Fischer 2002), Thompson sampling (Thompson 1933), Bayes-UCB (Kaufmann, Cappé, and Garivier 2012) and so on. The lower regret bound of these algorithms is $\Omega(\sqrt{NT})$ which will be not feasible when arm number N is large. Contextual bandit algorithms were then proposed to improve the efficiency when the arm number is large or infinite. Many linear bandit algorithms (Auer 2002; Abbasi-Yadkori, Pál, and Szepesvári 2011; Agrawal and Goyal 2013; Chu et al. 2011; Dani, Hayes, and Kakade 2008; Rusmevichientong and Tsitsiklis 2010) were proposed which assume the expected reward is a linear function of context feature. The best upper regret bound of these algorithms is $\tilde{O}(d\sqrt{T})$ which is free of arm number and efficient when the feature dimension d is small. Generalized linear bandit (Filippi et al. 2010; Li, Lu, and Zhou 2017) were also proposed. Besides, many highly non-linear bandit algorithms (Elmachtoub et al. 2017; Foster et al. 2018; Srinivas et al. 2012; Krause and Ong 2011; Valko et al. 2013) emerge in recent years where the expected reward is assumed a non-linear function of context feature, such as decision tree, Gaussian process. Although efficient in the environment with assumed reward function, the expected reward may deviate from the assumed reward function in practice because of function assumption error and/or incomplete context feature. Then, these contextual bandit algorithms may suffer linear regret bound in more general semi-parametric environment.

Few works study the stochastic multi-armed bandit problem in semi-parametric environment. As the works in (Gopalan, Maillard, and Zaki 2016; Besbes and Zeevi 2015) prove that the regret bound of linear bandit algorithms can still be sub-linear when the expected reward deviate slightly from linear function, the work in (Ghosh, Chowdhury, and Gopalan 2017) proposed a novel algorithm, called RLB, which can work in semi-parametric environment. It first distinguishes whether the bias is large or not, then applies linear bandit algorithm when bias is in the range that the regret bound of linear bandit algorithm is sub-linear and classic MAB algorithm is applied otherwise. Then, RLB will also be inefficient for large arm set. The work in (Krishnamurthy, Wu, and Syrgkanis 2018) just estimates the parametric part of expected reward and cannot be extended to general semi-parametric environment.

3 Problem Formulation

Suppose there are N arms, $\{1, 2, \cdots, N\}$. For arm i, $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector which is known, $r_i \in \mathbb{R}$ is the expected reward which is unknown. Denote $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and $\mathbf{r} = \{r_1, r_2, \cdots, r_N\}$. Expected reward can be formulated with a semi-parametric form

$$r_i = f(\theta^*, \mathbf{x}_i) + \epsilon_i \quad , \tag{1}$$

where $f(\theta^*, \mathbf{x}_i)$ is a parametric reward function of arm feature $\mathbf{x}_i$, θ^* is the optimal function parameter

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} |r_i - f(\theta, \mathbf{x}_i)| \quad , \tag{2}$$

and $\epsilon_i \in \mathbb{R}$ is the bias of parametric function f from real expected reward r_i. Note that the bias ϵ_i is independent among arms and with no parametric assumption. We call it the non-parametric part of expected reward. Denote $\epsilon_{max} = \max\{|\epsilon_1|, \cdots, |\epsilon_N|\}$ as the maximum bias. Assume $\|\theta^*\| \leq 1$ and $\|\mathbf{x}_i\| \leq 1$. For the linear parametric part case, $f(\theta^*, \mathbf{x}) = {\theta^*}^{\top}\mathbf{x}$. We call the environment with semi-parametric expected reward as semi-parametric environment. Actually, semi-parametric reward is a very general reward form. As it assumes a unique non-parametric part to each arm, it does not have strong assumption on parametric reward function.

Then, the agent faces a sequential decision making problem where the agent needs to maximize cumulative expected rewards. Formally, let T be the length of time horizon. At each time step t, the agent needs to select an arm i_t, then receives corresponding stochastic reward $\tilde{r}_t = r_{i_t} + \eta_t$ where

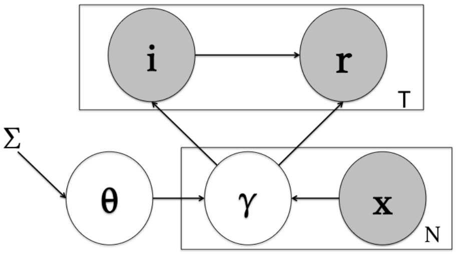

Figure 1: Graphical model of semi-parameteric sampling.

η_t is a zero-mean random noise (i.e. $\mathbb{E}(\eta_t) = 0$). $\{\eta_t\}$ are independent conditioned on i_t. Define the filtration as $\mathcal{F}_t = \{i_1, r_1, \cdots, i_t, r_t\} \cup \{\mathbf{X}\}$. The objective is to find a policy that can maximize the cumulative expected reward

$$\max \sum_{t=1}^{T} r_{i_t} \quad . \tag{3}$$

As expected rewards are unknown, the agent needs to explore arms to estimate expected reward and exploit arms with highest estimated reward at the same time. So, how to find an optimal tradeoff between exploration and exploitation is the key problem.

Since direct analysis of cumulative reward is not tractable, we analyze the cumulative regret instead

$$R(T, \mathbf{X}) = \sum_{t=1}^{T} (r_{i^*} - r_{i_t}) \quad , \tag{4}$$

which is the difference on reward between the optimal arm and the selected arm. Specifically, we adopt Bayesian regret, which is the expectation of regret over a prior distribution of $\mathbf{r}$ and θ^*.

$$\text{BayesRegret}(T, \mathbf{X}) = \mathbb{E}\left(R(T, \mathbf{X})\right) \quad . \tag{5}$$

Bayesian regret is a widely used metric of bandit method performance (Russo and Van Roy 2014). In practice, a feasible policy should achieve sub-linear cumulative regret with respect to the length of time horizon T which implies that the selected arm will converge to the optimal arm and the regret of one step will vanish.

Here, we define some notations used later. Let $\tau_{i,t} = \{l < t : i_l = i\}$ be the set of selected time steps of arm i before time step t, $n_{i,t} = |\{l < t : i_l = i\}|$ be the selected times, and $\overline{r}_{i,t} = \frac{\sum_{\{l<t:i_l=i\}} \tilde{r}_t}{n_{i,t}}$ be the empirical mean of stochastic rewards which will finally converge to real expected reward r_i.

4 Semi-Parametric Sampling

To achieve unbiased and efficient bandit algorithms, we propose a novel framework called *Semi-Parametric Sampling* (SPS) which is designed as a combination of parametric part and non-parametric part according to the definition of expected reward. SPS inherits the efficiency of parametric bandit and unbiased property of non-parametric bandit. The parametric part helps to provide a prior of expected reward to rapidly reduce the candidate arm set and avoid large regret. Thus the non-parametric part can work in a relatively small candidate set to efficiently estimate the expected reward and correctly identify the optimal arm .

Before going into the detail of the framework, we first give a brief introduction of Thompson sampling. Thompson sampling (Thompson 1933) is a practical design principle for bandit algorithms which selects arms by sampling from the posterior distribution of optimal arm on candidate arms. It assumes a prior distribution of optimal arm and the posterior distribution is conditioned on historical instances, including selected arms, rewards and observations. The posterior distribution will concentrate to the optimal arm as historical instances are collected.

In practice, instead of directly modeling the distribution of optimal arm, modeling the distribution of expected reward is adopted because of simpler implementation. In each time step, the agent will sample a reward from the posterior distribution of expected reward for each arm and select the arm with largest sampled reward. Such a sampling process is equivalent to direct sampling from posterior distribution of optimal arm.

Algorithm 1 Semi-Parametric Sampling

```
$t \leftarrow 1$
**repeat**
    update posterior probability of parameter $\theta$, $P(\theta_t|\mathcal{F}_{t-1})$, according to Equation (6)
    sample $\theta_t$ from $P(\theta_t|\mathcal{F}_{t-1})$
    **for** $i \in \{1, 2, \cdots, N\}$ **do**
        update posterior probability of expected reward, $P(\gamma_i|\mathcal{F}_{t-1}, \theta)$, according to Equation (7)
        sample $\gamma_{i,t}$ from $P(\gamma_i|\mathcal{F}_{t-1}, \theta)$
    **end for**
    select arm $i_t = \arg\max_i \gamma_{i,t}$ and observe reward $\tilde{r}_t$
    $t \leftarrow t + 1$
**until** $t > T$
```

SPS adopts a hierarchical generative process to model the generation of stochastic reward (see Figure 1).

- Step 1: The parameter of parametric part, θ, is drawn, and we can obtain the parametric part of reward by substituting θ and arm feature $\mathbf{x}$ into $f(\theta, \mathbf{x})$. Note that θ is shared by all the arms
- Step 2: As the expected reward deviates from parametric part, the expected reward, γ_i, is drawn from a distribution whose mean is the parametric part, $f(\theta, \mathbf{x}_i)$. The bias ϵ_i is modeled in this step, i.e. $\epsilon_i = \gamma_i - f(\theta, \mathbf{x}_i)$. So, the initialization of the variance of γ_i depends on the maximum bias, ϵ_{max}.
- Step 3: Stochastic reward, $\tilde{r}_t$, is drawn from a distribution whose mean is the expected reward, γ_{i_t}. That is $\eta_t = \tilde{r}_t - \gamma_{i_t}$ is zero-mean noise.

Algorithm 1 shows the process of SPS. The posterior distribution of θ can be formulated as

$$P\left(\theta|\mathcal{F}_{t-1}\right) \\ = \frac{1}{Z_{\theta,t-1}} \int_{\Gamma} \prod_{l=1}^{t-1} P\left(\tilde{r}_l|\gamma_{i_l}\right) \prod_{i=1}^{N} P\left(\gamma_i|f\left(\theta,\mathbf{x}_i\right)\right) P\left(\theta\right) d\Gamma \quad , \tag{6}$$

where $\Gamma = \{\gamma_1, \cdots, \gamma_N\}$, $Z_{\theta,t-1}$ is normalization factor. According to Equation (6), θ can aggregate the information of all the arms, including selected arms, returned stochastic reward and arm features. Thus, the distribution of θ will concentrate fast. Although there exists bias, $f(\theta, \mathbf{x})$ provides a low-variance base for expected reward and arms with small parametric part will be selected much less. As the arms with small parametric part are probably with small expected reward, less selection of these arms will lead to smaller regret.

The posterior distribution of expected reward $gamma_i$ is

$$P(\gamma_i|\mathcal{F}_{t-1}, \theta) \\ = \frac{1}{Z_{i,t-1}} \prod_{l \in \{\tau < t : i_\tau = i\}} P(\tilde{r}_l|\gamma_i) P\left(\gamma_i|f\left(\theta, \mathbf{x}_i\right)\right) \quad , \tag{7}$$

where $Z_{i,t-1}$ is normalization factor. Given $f\left(\theta, \mathbf{x}_i\right)$, expected rewards in Γ are estimated independently.

To implement the framework, the distribution of stochastic reward, $P(\tilde{r}_l|\gamma_i)$, the distribution of expected reward, $P\left(\gamma_i|f\left(\theta,\mathbf{x}_i\right)\right)$, and the prior distribution, $P\left(\theta\right)$, need to be specified.

4.1 Linear Parametric Part Case

We give an implementation of Semi-Parametric Sampling when the parametric part of expected reward is a linear function, that is $f(\theta, \mathbf{x}_i) = \theta^\top \mathbf{x}_i$. We call it *Linear Semi-Parametric Sampling* (LSPS). The distribution of stochastic reward, expected reward and linear parameter are all implemented by Gaussian distribution. Specifically,

$$\tilde{r}_t|\gamma_{i_t} \sim \mathcal{N}(\gamma_{i_t}, \sigma_1^2) \quad , \tag{8}$$

$$\gamma_i|\theta \sim \mathcal{N}(\theta^\top \mathbf{x}_i, \sigma_2^2) \quad , \tag{9}$$

$$\theta \sim \mathcal{N}(\mathbf{0}, \sigma_3^2 \mathbf{I}) \quad , \tag{10}$$

where σ_1, σ_2 and σ_3 are all hyper-parameters, $\mathbf{0}$ is a d-dimension vector whose elements are all zero and $\mathbf{I}$ is an d-by-d identity matrix. Substitute Equation (8), (9), (10) into Equation (6), we can obtain the posterior distribution of θ.

$$\theta|\mathcal{F}_{t-1} \sim \mathcal{N}(\hat{\theta}_t, \mathbf{A}_t^{-1}) \quad , \tag{11}$$

where

$$\mathbf{A}_t = \frac{1}{\sigma_3^2}\mathbf{I} + \sum_{i \in S} \frac{n_{i,t}}{\sigma_1^2 + n_{i,t}\sigma_2^2} \mathbf{x}_i \mathbf{x}_i^\top \quad , \tag{12}$$

$$\mathbf{b}_t = \frac{1}{\sigma_3^2}\mu + \sum_{i \in S} \frac{n_{i,t}\overline{r}_{i,t}}{\sigma_1^2 + n_{i,t}\sigma_2^2} \mathbf{x}_i \quad , \tag{13}$$

$$\hat{\theta}_t = \mathbf{A}_t^{-1}\mathbf{b}_t \quad , \tag{14}$$

Substitute Equation (8), (9), (10) into Equation (7) and assume θ_t be the sampled linear parameter from posterior distribution, we can obtain the posterior distribution of expected rewards Γ,

$$\gamma_i|\mathcal{F}_{t-1}, \theta_t \sim \mathcal{N}(\hat{\gamma}_{i,t}, \sigma_{i,t}^2) \quad , \tag{15}$$

where

$$\hat{\gamma}_{i,t} = \frac{\sigma_2^2 n_{i,t}\overline{r}_{i,t} + \sigma_1^2 \theta_t^\top \mathbf{x}_i}{\sigma_1^2 + n_{i,t}\sigma_2^2} \quad , \tag{16}$$

$$\sigma_{i,t}^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + n_{i,t}\sigma_2^2} \quad . \tag{17}$$

The empirical estimation of expected reward $\hat{\gamma}_{i,t}$ is a weighted average between non-parametric empirical mean $\overline{r}_{i,t}$ and parametric function value $\theta_t^\top \mathbf{x}_i$. When selected times $n_{i,t}$ is small, the parametric part $\theta_t^\top \mathbf{x}_i$ dominates the estimation. With $n_{i,t}$ increasing, the non-parametric part will dominates the estimation and the variance $\delta_{i,t}$ will decrease which means that the distribution of γ_i will concentrate to the real expected reward. Algorithm 2 describes the update and sampling process in detail.

Algorithm 2 Linear Semi-Parametric Sampling

```
$S \leftarrow \emptyset, t \leftarrow 1$
repeat
    $\mathbf{A}_t \leftarrow \frac{1}{\sigma_3^2}\mathbf{I} + \sum_{i \in S} \frac{n_{i,t}}{\sigma_1^2 + n_{i,t}\sigma_2^2}\mathbf{x}_i\mathbf{x}_i^\top$
    $\mathbf{b}_t \leftarrow \frac{1}{\sigma_3^2}\mu + \sum_{i \in S} \frac{n_{i,t}\overline{r}_{i,t}}{\sigma_1^2 + n_{i,t}\sigma_2^2}\mathbf{x}_i$
    sample $\theta_t$ from $\mathcal{N}(\mathbf{A}_t^{-1}\mathbf{b}_t, \mathbf{A}_t^{-1})$
    for $i \in \{1, 2, \cdots, N\}$ do
        sample $\gamma_{i,t}$ from $\mathcal{N}(\hat{\gamma}_{i,t}, \sigma_{i,t}^2)$
    end for
    select arm $i_t = \arg\max_i \gamma_{i,t}$ and observe reward $\tilde{r}_t$
    $S \leftarrow S \cup \{i_t\}$
    $t \leftarrow t + 1$
until $t > T$
```

Note that Thompson sampling with linear payoff (Agrawal and Goyal 2013) and non-parametric Thompson sampling with Gaussian priors (Agrawal and Goyal 2017) are both special cases of Algorithm 2. If $\sigma_2 = 0$, then the algorithm becomes Thompson sampling with linear payoff. And if $\sigma_2 \rightarrow \infty$, then the algorithm becomes non-parametric Thompson sampling.

Moreover, prior knowledge of bias can help us tune the hyper-parameters to achieve lower regret. The variance σ_2^2 is closely related to the bias ϵ_{max}. Intuitively, σ_2^2 determines the size of the region around parametric part to explore. If the bias ϵ_{max} is small, we only need to explore a small region around the parametric part and can set a relatively small σ_2. Then the variance $\sigma_{i,t}^2$ will be small and the parametric part will make major contribution to the sampling of expected reward. Finally, the algorithm is expected to achieve regret bound close to Thompson sampling with linear payoff (i.e. $\tilde{O}(d\sqrt{T})$). So, when the bias is small, LSPS can achieve much lower regret bound than non-parametric Thompson sampling when arm feature dimension is much smaller than arm number. On the other hand, if the bias is large, σ_2 should also be large enough to guarantee that the real expected reward can be explored. Then, the algorithm is close to non-parametric Thompson sampling and will achieve regret bound close to $\tilde{O}(\sqrt{NT})$.

The variance σ_1^2 is closely related to the variance of stochastic reward. If the variance of stochastic reward is large, σ_1 should also be large to prevent that the posterior distribution of expected reward concentrates too fast to wrong value and optimal arm is missed. When σ_1 is large, the variance $\sigma_{i,t}^2$ and $\mathbf{A}_t$ will also be large. Then the posterior distribution needs more instances to concentrate. On the other hand, if the variance of stochastic reward is small, σ_1^2 can be relatively small which will make the posterior distribution concentrate fast and achieve lower regret.

In summary, the hyper-parameters, σ_1 and σ_2, control the balance between exploration and exploitation. Too large hyper-parameters will lead to more exploration and larger regret, but the algorithm will still find the optimal arm and regret in one time step will vanish. Too small hyper-parameters will increase the probability that the algorithm converges to a sub-optimal arm and the regret in one time step will never vanish. Based on above intuitive analysis, we give a formal regret analysis in next section.

5 Regret Analysis

According to Proposition 3 (Russo and Van Roy 2016), the bayesian regret bound of Thompson sampling with finite arms is not worse than $\tilde{O}(\sqrt{NT})$ which is also a upper bayesian regret bound of SPS framework.

In this section, we mainly focus on the regret analysis of LSPS and attempt to give a tighter bayesian regret bound. We first give the analysis result

Theorem 1. *If $\forall t \le T$, η_t is R-sub-Gaussian, i.e.*

$$\mathbb{E}\left(e^{\lambda\eta_t}\right) \le e^{\lambda^2 R^2/2} \quad , \quad \forall \lambda \ge 0 \quad ,$$

$d \le \sqrt{N}$ and $\epsilon_{max} \le \sqrt{\frac{N}{dT}\left(\frac{d}{\sqrt{N}}\right)^{2\alpha}}$ where $\alpha \in [0,1]$, then, the algorithm LSPS can achieve upper bayesian regret bound

$$\tilde{O}\left(\sqrt{N}^{1-\alpha} d^{\alpha} \sqrt{T}\right) \tag{18}$$

with

$$\frac{\sigma_2^2}{\sigma_1^2} = \frac{T}{N}\left(\frac{d}{\sqrt{N}}\right)^{\alpha} \quad . \tag{19}$$

In semi-parametric environment, compared to existing bandit algorithms, LSPS achieves at least $\left(\sqrt{N}/d\right)^{\alpha}$-order improvement. If $d \ll \sqrt{N}$, the improvement will be significant. According to the bound of bias ϵ_{max}, smaller bias can lead to larger α which means larger improvement. Moreover, if N is comparable to T, then LSPS can achieve significant improvement even when the bias is relatively large.

We will give the proof sketch below. The proof consists of two steps. In the first step, we prove the high probability bound of difference between the sampled expected reward and the real expected reward where the bound will converge to zero. This means the sampled expected reward will converge to real expected reward. In the second step, with the result of first step, we can construct the Upper Confidence Bound (UCB) of expected reward. Then, according to Proposition 1, the regret can be decomposed into the cumulative confidence interval of the selected arms. We prove the bound of parametric part and non-parametric part respectively, then sum them up as the final regret bound.

Proposition 1. *(Proposition 1 in (Russo and Van Roy 2014)) For any upper confidence bound function sequence $\{U_t | t \le T\}$, for all $T \in \mathbb{N}$,*

$$\text{BayesRegret}\,(T, \mathbf{X}) = \mathbb{E}\left(U_t(i_t) - r_{i_t}\right) + \mathbb{E}\left(r_{i^*} - U_t(i^*)\right).$$

5.1 High Probability Bound of Sampled Expected Reward

The difference between the sampled expected reward and the real expected reward can be decomposed into four parts by backtracking the sampling process of the expected reward.

$$\begin{aligned}
|\gamma_{i,t} - r_i| \le& |\gamma_{i,t} - \hat{\gamma}_{i,t}| + |\hat{\gamma}_{i,t} - r_i| \\
|\hat{\gamma}_{i,t} - r_i| \le& \frac{n_{i,t}\sigma_2^2}{\sigma_1^2 + n_{i,t}\sigma_2^2}|\overline{r}_{i,t} - r_i| \\
&+ \frac{\sigma_1^2}{\sigma_1^2 + n_{i,t}\sigma_2^2}|\tilde{\theta}_t^\top \mathbf{x}_i - \hat{\theta}_t^\top \mathbf{x}_i| \\
&+ \frac{\sigma_1^2}{\sigma_1^2 + n_{i,t}\sigma_2^2}|\hat{\theta}_t^\top \mathbf{x}_i - r_i|
\end{aligned}$$

The difference is first decomposed into the difference between sampled expected reward $\gamma_{i,t}$ and estimated expected reward $|\gamma_{i,t} - \hat{\gamma}_{i,t}|$ and the difference between estimated expected reward and real expected reward $|\hat{\gamma}_{i,t} - r_i|$. $|\hat{\gamma}_{i,t} - r_i|$ can be decomposed into three parts according to the formulation of $\hat{\gamma}_{i,t}$. The high probability bound of the four parts are given below respectively.

Lemma 1 is to bound the difference between sampled expected reward $\gamma_{i,t}$ and estimated expected reward $\hat{\gamma}_{i,t}$.

Lemma 1. *For any $t \le T$, with probability $1 - \frac{\delta}{NT^2}$,*

$$|\gamma_{i,t} - \hat{\gamma}_{i,t}| \le \sqrt{2\ln\frac{NT^2}{2\delta}}\sigma_{i,t} \tag{20}$$

Proof. According to Formula 7.1.13 from (Abramowitz and Stegun 1965),

$$P\left(|\gamma_{i,t} - \hat{\gamma}_{i,t}| \ge a \cdot \sigma_{i,t}\right) \le \frac{1}{2}e^{-a^2/2} \quad .$$

Let $\frac{\delta}{NT^2} = \frac{1}{2}e^{-a^2/2}$, then we can prove the lemma. □

Lemma 2 is to bound the difference between empirical mean of expected reward and real expected reward.

Lemma 2. *For any $t \le T$ and $i \le N$, if η_t is R-sub-Gaussian, i.e.*

$$\mathbb{E}\left(e^{\lambda\eta_t}\right) \le e^{\lambda^2 R^2/2} \quad , \quad \forall \lambda \ge 0 \quad , \tag{21}$$

then, with probability $1 - \frac{\delta}{NT^2}$,

$$|\overline{r}_{i,t} - r_i| \le R\sqrt{\frac{2\ln 2NT^2/\delta}{n_{i,t}}} \quad . \tag{22}$$

It is easy to prove the lemma by applying the Azuma-Hoeffding inequality.

Lemma 3 is to bound the difference between sampled parametric part and estimated parametric part.

Lemma 3. *For any $t \leq T$ and $i \leq N$, with probability $1 - \frac{\delta}{NT^2}$,*

$$|\tilde{\theta}_t^\top \mathbf{x}_i - \hat{\theta}_t^\top \mathbf{x}_i| \leq \sqrt{2d \ln \frac{dNT^2}{2\delta}} \sqrt{\mathbf{x}_i^\top \mathbf{A}_t^{-1} \mathbf{x}_i} \quad . \tag{23}$$

The lemma can be proven by applying Cauchy-Schwarz inequality and Formula 7.1.13 from (Abramowitz and Stegun 1965).

Lemma 4 is to bound the difference between estimated parametric part and real expected reward. Because of the bias of parametric function, the difference will not vanish. However, as the weight of parametric part will vanish with the selected times increasing, the total difference will still converge to zero.

Lemma 4. *For any $t \leq T$ and $i \leq N$, if η_t is R-sub-Gaussian, i.e.*

$$\mathbb{E}\left(e^{\lambda \eta_t}\right) \leq e^{\lambda^2 R^2/2} \quad , \quad \forall \lambda \geq 0 \quad , \tag{24}$$

then, with probability $1 - \frac{\delta}{NT^2}$,

$$\begin{aligned} |\hat{\theta}_t^\top \mathbf{x}_i - r_i| \leq & \epsilon_{max} + \sqrt{\mathbf{x}_i^\top \mathbf{A}_t^{-1} \mathbf{x}_i} \sqrt{\frac{Nt}{N\sigma_1^2 + t\sigma_2^2}} \epsilon_{max} \\ & + 2R\sqrt{2d \ln \frac{NT^2}{2\delta}} \sqrt{\mathbf{x}_i^\top \mathbf{A}_t^{-1} \mathbf{x}_i} \quad . \end{aligned} \tag{25}$$

Combine the results in Lemma 1, Lemma 2, Lemma 3 and Lemma 4, we can get the high probability bound of sampled expected reward.

Lemma 5. *For any $t \leq T$ and $i \leq N$, with probability $1 - \frac{4\delta}{NT^2}$,*

$$|\gamma_{i,t} - r_i| \leq g_{i,t},$$

where

$$\begin{aligned} g_{i,t} = & \sqrt{2 \ln \frac{NT^2}{2\delta}} \sqrt{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + n_{i,t}\sigma_2^2}} \\ & + \frac{n_{i,t}\sigma_2^2}{\sigma_1^2 + n_{i,t}\sigma_2^2} \cdot R\sqrt{\frac{2 \ln 2NT^2/\delta}{n_{i,t}}} \\ & + \frac{\sigma_1^2}{\sigma_1^2 + n_{i,t}\sigma_2^2} (2R\sqrt{2d \ln \frac{NT^2}{2\delta}} \sqrt{\mathbf{x}_i^\top \mathbf{A}_t^{-1} \mathbf{x}_i} \\ & + \epsilon_{max} + \sqrt{\mathbf{x}_i^\top \mathbf{A}_t^{-1} \mathbf{x}_i} \sqrt{\frac{Nt}{N\sigma_1^2 + t\sigma_2^2}} \epsilon_{max}) \quad . \end{aligned} \tag{26}$$

5.2 Bayesian Regret Bound

With Lemma 5, we can construct the UCB of expected reward as $U_t(i) = \gamma_{i,t} + g_{i,t}$. Then, we can exploit Proposition 1

$$\begin{aligned} \mathbb{E}(\sum_t \Delta_{i_t}) &= \mathbb{E}(\sum_t r_{i^*} - r_{i^*}^{UCB}) + \mathbb{E}(\sum_t r_{i_t}^{UCB} - r_{i_t}) \\ &\leq \frac{\delta}{T} + \mathbb{E}(2 \sum_t g_{i_t,t}) \quad . \end{aligned}$$

Finally, we have the bayesian regret bound described in Lemma 6.

Lemma 6. *For any $t \leq T$ and $i \leq N$, if η_t is R-sub-Gaussian, i.e.*

$$\mathbb{E}\left(e^{\lambda \eta_t}\right) \leq e^{\lambda^2 R^2/2} \quad , \quad \forall \lambda \geq 0 \quad , \tag{27}$$

then, the bayesian regret bound of LSPS is

$$\begin{aligned} \mathbb{E}(\sum_t \Delta_{i_t}) \leq & \frac{\delta}{T} + R \cdot \frac{\sigma_2^2 T}{\sigma_1^2 N + \sigma_2^2 T} \sqrt{2NT \ln 2T^2/\delta} \\ & + 3\sigma_1 R d \sqrt{2T \ln \frac{dT^2}{2\delta} \ln T} \\ & + 11 \frac{\sigma_1^2}{\sigma_2^2} \ln \left(\sigma_1^2 + \frac{T}{N}\sigma_2^2\right) N\sqrt{d}\epsilon_{max} \quad . \end{aligned} \tag{28}$$

The detailed proof is in supplementary materials. Theorem 1 can be proven by simply applying Lemma 6.

6 Experiment

We conduct experiments on synthetic data and an e-commerce dataset in different environments to evaluate the performance of LSPS.

6.1 Settings

Three alternative algorithms, including representative bandit algorithms, including both classic MAB algorithm and contextual bandit algorithm, are compared:

- TS-Gau (Agrawal and Goyal 2017): This algorithm is a classic MAB algorithm which adopts Thompson sampling with Gaussian prior. It is a special case of LSPS with $\sigma_2 \to \infty$. Other hyper-parameters are set the same value as LSPS.
- TS-Beta (Agrawal and Goyal 2017): This algorithm is a classic MAB bandit algorithm which adopts Thompson sampling with Beta prior. It is designed for binary stochastic reward, i.e. $\tilde{r}_t \in \{0, 1\}$
- TS-Lin (Agrawal and Goyal 2013): This algorithm is a linear bandit algorithm which adopts Thompson sampling with Gaussian prior. It is also a special case of LSPS with $\sigma_2 = 0$. Other hyper-parameters are set the same value as LSPS.

The synthetic data is randomly generated. We first sample context feature and linear parameter from standard Gaussian distribution, and all the sampled data is transformed to

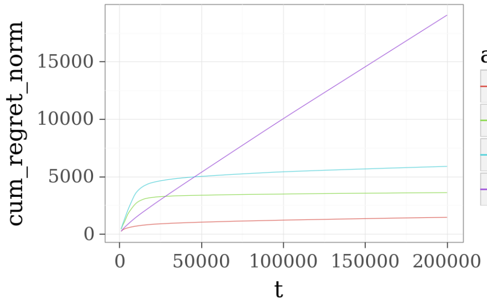

Figure 2: Cumulative regret in semi-parametric environment with $a = 0.5$.

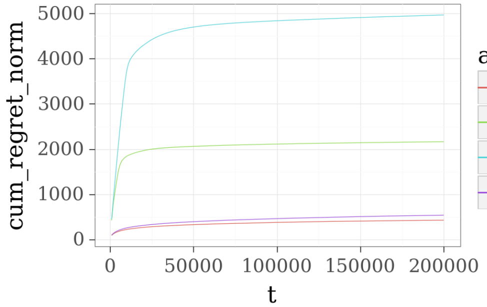

Figure 3: Cumulative regret in linear environment.

its absolute value. Then context feature vectors are normalized by the same factor so that the largest norm equals to 1. The linear parameter vector is normalized so that its norm equals to a predefined constant $a \in [0, 1]$. Finally, the bias of each arm is uniformly sampled from $[1 - a, 1]$. In each time step, stochastic reward of an arm is sampled from a Gaussian distribution or binomial distribution whose expectation is the expected reward of the arm.

The e-commerce dataset is collected from an online e-commerce platform. It contains 1000 items with 5-dimension feature. The expected reward function is a DNN model trained on the historical user behavior log on the items. In each time step, stochastic reward of an arm is sampled from a binomial distribution whose expectation is the expected reward of the arm.

6.2 Results

Figure 2 and Figure 3 show the results on synthetic data with 1000 arms and 5-dimension feature. Figure 2 shows that LSPS achieves much lower regret than the others in semi-parametric environment. Note that even when the bias is large ($a = 0.5$), LSPS still outperforms classic MAB algorithms. This is because the classic MAB algorithms ignore the correlation on feature and need to explore more than LSPS. Moreover, as the expected reward deviate a lot from the expected reward, the regret of TS-Lin is nearly linear. Figure 3 shows the result in linear environment. LSPS can achieve comparable regret with linear bandit algorithm (TS-Lin).

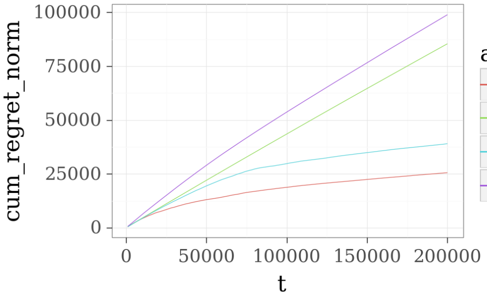

Figure 4: Cumulative regret in the e-commerce dataset.

Figure 4 shows the results in the e-commerce dataset. As linear reward function is a mis-specified function, TS-Lin suffers large regret. With the help of non-parametric part, LSPS achieves much lower regret.

7 Conclusion

We propose a novel framework, called *Semi-Parametric Sampling* (SPS), which can work in the semi-parametric environment and is expected to receive smaller regret than existing bandit algorithms. An implementation of SPS is given when the parametric part of expected reward is linear function. And we prove better regret bound than linear bandit and classic MAB algorithms. The experiments also demonstrate that the regret of LSPS is smaller. In the future, we will focus on automatic identification of the bias from parametric part in order to automatically set the hyper-parameters and obtain largest cumulative reward.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Abramowitz, M., and Stegun, I. A. 1965. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation.

Agrawal, S., and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

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

Agrawal, R. 1995. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* 27(4):1054–1078.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3(Nov):397–422.

Besbes, O., and Zeevi, A. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science* 61(4):723–739.

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

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 355–366.

Elmachtoub, A. N.; McNellis, R.; Oh, S.; and Petrik, M. 2017. A practical method for solving contextual bandit problems using decision trees. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, 586–594.

Foster, D. J.; Agarwal, A.; Dudík, M.; Luo, H.; and Schapire, R. E. 2018. Practical contextual bandits with regression oracles. *Proceedings of the 35th International Conference on Machine Learning* 1534–1543.

Ghosh, A.; Chowdhury, S. R.; and Gopalan, A. 2017. Misspecified linear bandits. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 3761–3767.

Gopalan, A.; Maillard, O.; and Zaki, M. 2016. Low-rank bandits with latent mixtures. *CoRR* abs/1609.01508.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2012. On bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*, 592–600.

Krause, A., and Ong, C. S. 2011. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, 2447–2455.

Krishnamurthy, A.; Wu, Z. S.; and Syrgkanis, V. 2018. Semiparametric contextual bandits. *International Conference on Machine Learning* 80:2776–2785.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably optimal algorithms for generalized linear contextual bandits. *International Conference on Machine Learning* 70:2071–2080.

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

Russo, D., and Van Roy, B. 2014. Learning to optimize via posterior sampling. *Mathematics of Operations Research* 39(4):1221–1243.

Russo, D., and Van Roy, B. 2016. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research* 17(1):2442–2471.

Schwartz, E. M.; Bradlow, E. T.; and Fader, P. S. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* 36(4):500–522.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. W. 2012. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Information Theory* 58(5):3250–3265.

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

Valko, M.; Korda, N.; Munos, R.; Flaounas, I.; and Cristianini, N. 2013. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*.

Wang, L.; Zhang, W.; He, X.; and Zha, H. 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2447–2456. ACM.