(Nearly) Optimal Differentially Private Stochastic Multi-Arm Bandits

Nikita Mishra
Department of Computer Science
University of Chicago
nmishra@cs.uchicago.edu

Abhradeep Thakurta
Yahoo! Research, Sunnyvale
guhathakurta.abhradeep@gmail.com

Abstract

We study the problem of private stochastic multi-arm bandits. Our notion of privacy is the same as some of the earlier works in the general area of private online learning [13, 17, 24]. We design algorithms that are i) differentially private, and ii) have regret guarantees that (almost) match the regret guarantees for the best non-private algorithms (e.g., upper confidence bound sampling and Thompson sampling). Moreover, through our experiments, we empirically show the effectiveness of our algorithms.

1 INTRODUCTION

A general abstraction of bandit problems is the following: Given a set of k arms $\mathcal{C} = \{a_1, \cdots, a_k\}$, at each time step one pulls an arm $a_i \in \mathcal{C}$, gets a reward corresponding to a_i and the objective of the algorithm is to obtain large cumulative reward over all time steps T. In a class of problems called the *adversarial bandits*, it is assumed that the reward can be adversarial, in the sense that the rewards for each of the arms in $\mathcal{C}$ are arbitrarily chosen. The other class of problems is called the *stochastic bandits* where it is assumed that the reward for each of the arm in $\mathcal{C}$ is from an unknown distribution. (See [5] for a detailed introduction to these classes.) In this paper we are interested in the stochastic bandit setting. We analyze two of the most common algorithms in this setting in the context of differential privacy, *upper confidence bound* (UCB) sampling by [4] and *Thompson sampling* by [2, 18]. We show that one can modify UCB sampling and Thompson sampling algorithms in such a way that ensures: i) differential privacy, and ii) regret guarantee which is only poly $\log T$ factor worse compared to the regret for the non-private variants.

We now focus on the semantics of differential privacy in this setting where the data points (the rewards) arrive online in a stream at every time step. This setting was first studied by [13] and then followed by [17] and [24]. Let $f_t = \langle f_t(a_1), \cdots, f_t(a_k) \rangle$ be the vector of rewards for all the arms in $\mathcal{C}$ at time step t. Privacy guarantee will ensure that from the output of the algorithm over all the T time steps the adversary will not be able to distinguish between the presence or absence of any single reward vector f_t. [16] studied differentially private online algorithms in the *full-information* setting, where at each time step t the algorithm can see the complete reward vector f_t as opposed to $f_t(a)$ for the arm a pulled in the bandit setting. [24] extended this line of work to obtain tighter and nearly optimal regret guarantees for both full-information and adversarial bandit settings. Recall that in the non-private world, the full-information and the adversarial bandit settings both have optimal regret guarantee of $\Omega(\sqrt{T})$ (see [23]). In contrast, stochastic bandit algorithms enjoy a regret of $O(\log T)$. In this work we obtain the *first* and *nearly optimal* regret guarantees for stochastic bandit problems. Since, stochastic bandit algorithms have a very different flavor than adversarial online algorithms, we needed to introduce new proof techniques tailored to our problem.

The stochastic multi-armed bandit algorithms usually run in the two implicit phases *exploration phase* and *exploitation phase*. During the exploration phase, the algorithm uses the pull of the arms in the initial rounds to get a sufficiently accurate estimate of the means of the arms, and then in the second phase uses this information to guide the decision of pulling of arms in the later rounds. However, in order to ensure differential privacy, we are required to introduce certain randomness in the observed rewards, but this tends to grossly corrupt the estimation of the means of the arms. At a high-level, for both UCB and Thompson sampling, we address this issue by increasing the number of rounds used by the algorithm to estimate these means. The exact details are very different for both the algorithms are discussed in the respective sections. One important point to keep in mind is that although we make stochastic assumptions on the data to ensure strong utility guarantees, we *do not* make any assumptions on the data while ensuring privacy for our algorithms. Using the distributional assumption on the data for any kind of privacy guarantee may be disastrous, since real world data may not fol-

low the assumed distribution. For our algorithms, privacy should hold in the worst case scenario but the utility guarantee holds under distributional assumptions on the data. Finally, to fortify our theoretical guarantees we show that they work well in experiments and often are competitive w.r.t. the non-private algorithms.

Practical motivation. In the past few years bandit algorithms have become extremely popular in both the theoretical and applied online learning community. Along with having strong theoretical guarantees, these algorithms have found wide applicability in Internet scale systems. A concrete usage scenario is in the online search advertisement industry. Companies like Google, Microsoft and Yahoo run multi-million dollar industry based on showing relevant advertisements for a web-queries. For a given search query by the user, the search engine displays a few advertisement which the user is most likely to click (commonly known as the *main-line advertisement*). After the advertisement gets displayed, the user chooses to either click or not click. If he/she clicks, then we say that the search engine gets a reward of one, and zero otherwise. To earn maximum revenue, the search engine strives to get high overall reward. One important feature of this setup (and generally in bandit learning) is that the search engine only gets to see the reward for the advertisements that were shown to the user, and gets no feedback about what the user would have done on other candidate advertisements. Bandit algorithms have been extremely successful in such settings and often enjoy strong theoretical guarantees for the overall reward. Settings like search advertisements immediately raise privacy concern for the users whose data get used in training the learning algorithms. Consider the example of a user who searches for a lawyer in Port Jefferson, NY and clicks on an advertisement of a divorce law farm. If the learning algorithm uses this feedback to show the same advertisement for similar query (e.g., finance lawyer in Port Jefferson, NY), then one can use such feedbacks to infer significant amount of information about a particular user. [19] on the Facebook advertisement recommendation system and [6] on the Amazon recommendation system, showed that one can leverage such side information to breach user privacy.

1.1 Our Contributions

Here, we provide an overview of our contributions.

- ***Differentially private UCB sampling [Section 3].*** We provide a differentially private variant of UCB sampling which enjoys the same utility guarantee as the non-private algorithm up to poly-logarithmic factors in the number of time steps T. The privacy guarantee follows via standard reduction to the *tree-based-aggregation scheme*, proposed by [14, 7]. Our utility analysis goes via carefully analyzing the *exploration phase* of the algorithm, where it estimates the means of the arms. Thus, we provide a version of UCB sampling algorithm which is robust to noise.
- ***Differentially private Thompson sampling [Section 4].*** We also provide a differentially private variant of Thompson sampling which enjoys the same utility guarantee as the non-private algorithm up to poly-logarithmic factors in the number of time steps T. The privacy analysis is very similar to the UCB sampling. But, the utility analysis is much trickier. In fact even without privacy analyzing the *exploration phase* was considered extremely difficult and it took more than seventy years for the community to come up with a formal analysis (see [2, 18]). The main technical contribution of this section is to come up with a noise robust version of Thompson sampling. In order to obtain this robustness, the exploration phase of the original algorithm had to be modified significantly. A question that we leave open in this work is that if that is necessary.

2 BACKGROUND AND PROBLEM DEFINITION

2.1 Background on Differential privacy

In this section we provide a short overview of differential privacy. $\mathcal{D} = \langle f_1, \cdots, f_T \rangle$ be a data set of all the reward functions. We call a data set $\mathcal{D}'$ neighbor of $\mathcal{D}$ if it differs from $\mathcal{D}$ in exactly one reward function. Let $\mathcal{C}^T$ be the space of all T outputs of Algorithm $\mathcal{A}$.

Definition 1 (Differential privacy [12])**.** *A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for any two neighboring data sets $\mathcal{D}$ and $\mathcal{D}'$, and for all sets $\mathcal{O} \subseteq \mathcal{C}^T$ the following holds:*

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{O}] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{O}].$$

As per the semantics of the definition, differential privacy ensures that an adversary gets to know *"almost the same thing"* about a reward function f_t irrespective of its presence or absence in the data set $\mathcal{D}$. This closeness is measured by the privacy parameter ϵ. A typical choice of ϵ is a small constant (e.g., 0.1). One important requirement of the definition is that the guarantee should hold for every pair of neighboring data sets. This directly implies that although for the regret analysis of our algorithm $\mathcal{A}$ we can assume that the reward function comes from some underlying distribution, but we *cannot* use any stochastic assumption on the reward functions for privacy guarantee. Next, we discuss some of the basic tools for designing differentially private algorithms.

Laplace and Gamma mechanism. Laplace [12] and Gamma mechanism [10] are sensitivity based methods to achieve differential privacy. The best way to introduce Laplace mechanism is via the following setting. Consider a domain of data entries $\mathcal{U}$ and a function $f : \mathcal{U}^* \to \mathbb{R}$. For the domain of data sets $\mathcal{U}^n$, we define the sensitivity of the

function f as below.

$$s = \mathsf{Sensitivity}(f) = \max_{\text{Neighbors}\mathcal{D},\mathcal{D}' \in \mathcal{U}^*} |f(\mathcal{D}) - f(\mathcal{D}')|$$

Let $\mathsf{Lap}(\lambda)$ be the Laplace distribution with scaling parameter λ, *i.e.,* the density function of this distribution is given by $\frac{1}{2\lambda} e^{-|x|/\lambda}$. Laplace mechanism states that for a given data set $\mathcal{D}$ and noise $N \sim \mathsf{Lap}\left(\frac{s}{\epsilon}\right)$, $f(\mathcal{D}) + N$ is ϵ-differentially private. The proof of this claim directly follows from the density function for Laplace distribution and triangle inequality. [12]

Gamma mechanism is also very similar to Laplace mechanism. The only difference being that we are now working with a vector valued function $f : \mathcal{U}^* \to \mathbb{R}^p$. Analogous to Laplace mechanism, let us define the L_2-sensitivity of the function f as below.

$$s = \mathsf{Sensitivity}(f) = \max_{\text{Neighbors}\mathcal{D},\mathcal{D}' \in \mathcal{U}^*} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$$

Gamma mechanism states that if we sample the noise vector $N \in \mathbb{R}^p$ from the noise distribution with kernel $e^{-\epsilon \|N\|_2 / s}$, then $f(\mathcal{D}) + N$ is ϵ-differentially private. (See [10] for the proof.)

Tree based aggregation. Initially proposed by [14, 7], this aggregation scheme is extremely effective in releasing private continual statistics over a data stream. To define the scheme formally, consider a data set $\mathcal{D} = \langle f_1, \cdots, f_T \rangle$, where each entry $f_t \in [0, 1]$, and these entries arrive in a stream, i.e., at every time step $t \in [T]$, one entry f_t arrives. At every time step t, the task is to output $v_t = \sum_{\tau=1}^{t} f_t$ while ensuring that the complete output sequence $\langle v_1 \cdots, v_T \rangle$ is ϵ-differentially private. One can design an algorithm (using a binary tree based aggregation), which assures an additive error of $O\left(\frac{\log^{1.5} T}{\epsilon}\right)$ per query. Moreover, it is simple to extend this scheme to the case where $f_t \in \mathbb{R}^p$ and $\|f_t\|_2 \leq 1$ for all $t \in [T]$. Since, the noise used in this scheme is exponential in nature, a high-probability guarantee is also immediate. We defer the details of the scheme to Appendix A. Suppose at every time step $t \in [T]$, one entry from dataset $\mathcal{D}$, $f_t \in [0, 1]$ arrives and the task is to output $v_t = \sum_{\tau=1}^{t} f_t$ while ensuring that the complete output sequence $\langle v_1 \cdots, v_T \rangle$ is ϵ-differentially private. This algorithm uses a binary tree based aggregation scheme, which assures an additive error of $O\left(\frac{\log^{1.5} T}{\epsilon}\right)$ per query. We defer the details of the scheme to Appendix A. Moreover, it can be extended to the case where $f_t \in \mathbb{R}^p$ and $\|f_t\|_2 \leq 1$ for all $t \in [T]$.

2.2 Background on Stochastic Multi-arm Bandits

A typical setup for an online learning problem is as follows: There is a sequence of *reward functions* $f_1, \cdots, f_T$ arriving in a stream (*i.e.,* one at every time step $t \in [T]$), where each f_i maps from some fixed set $\mathcal{C}$ to $\mathbb{R}$. At every time step t, an online learning algorithm $\mathcal{A}$ is expected to produce an element $x_t \in \mathcal{C}$ before f_t is revealed to it. Once f_t gets revealed to $\mathcal{A}$, the algorithm pays a reward of $f_t(x_t)$. The objective of $\mathcal{A}$ is to be competitive with the best choice of $x \in \mathcal{C}$ in the hindsight, *i.e.,* be competitive with $\max_{x \in \mathcal{C}} \sum_{t=1}^{T} f_t(x)$. A natural measure of the utility of $\mathcal{A}$ is *regret*, defined: $\mathsf{Regret}_{\mathcal{A}}(T) = \max_{x \in \mathcal{C}} \sum_{t=1}^{T} f_t(x) - \sum_{t=1}^{T} f_t(x_t)$. (For a detailed discussion, see [23]

There are two popular settings under which these problems are studied, namely, i) *online learning under complete feedback or the full-information setting*, and ii) *online learning under partial feedback or the bandit setting*. In the first setting, it is assumed that at time step t after the algorithm $\mathcal{A}$ has produced x_t, it gets to see the complete reward function f_t. In the second setting, the algorithm gets much lesser information from the environment and just sees the evaluation of f_t at x_t.

2.3 Problem Definition

Let us assume that for all $t \in [T]$ we have $f_t : \mathcal{C} \to [0, 1]$, where $\mathcal{C}$ is the set of k-arms. Additionally we assume that for each arm $a \in \mathcal{C}$, each $f_t(a)$ is an independent sample from a distribution with mean μ_a. The objective is to design differentially private algorithms, whose regret (defined in (1)) depends poly-logarithmically in the number of reward functions T.

$$\mathbb{E}\left[\mathsf{Regret}_{\mathcal{A}}(T)\right] = T \max_{a \in \mathcal{C}} \mu_a - \mathbb{E}\left[\sum_{t=1}^{T} f_t(a(t))\right]. \quad (1)$$

Here, $a(t) \in \mathcal{C}$ is the arm played in the t-th time step.

3 PRIVATE UCB SAMPLING

Upper Confidence Bound (UCB) sampling by [4] is a heuristic for stochastic multi-arm bandit (MAB) problems, which despite being very simple gives strong utility guarantees. The regret for UCB $O^*(\log T)$ in fact matches the asymptotic lower given by [20] upto a problem dependent constant. This is in sharp contrast with the algorithms for adversarial multi-arm bandit problems where the regret depends polynomially on the time horizon T (see [1, 15]). Recently [24] provided differentially private algorithms for adversarial bandit problems, which are almost optimal in the parameter T. In this section, for the UCB algorithm for stochastic MAB, we provide a differentially private algorithm whose expected regret is only poly-logarithmically worse in T. Before we move on to the differentially private UCB algorithm, we provide a brief overview of the non-private version of the algorithm.

Background on UCB sampling. Recall that in the MAB problem there are k-arms denoted by the set $\mathcal{C}$, and at each time step t each arm $a \in \mathcal{C}$ produces either 0 or 1 from some unknown but fixed distribution on $[0,1]$ with mean μ_a. The objective is to minimize the regret defined in (1). For each arm a, the UCB algorithm records the number of times it got pulled $n_a(t)$ and the average reward $\frac{r_a(t)}{n_a(t)}$ aggregated so far upto time t. Upon initialization, the algorithm pulls each arm exactly once. Later, the algorithm picks the arm with the highest upper confidence bound, i.e., $\arg\max_{a\in\mathcal{C}} \frac{r_a(t)}{n_a(t)} + \sqrt{\frac{2\log t}{n_a(t)}}$.

Theorem 2 (Regret for non-private UCB Sampling [4])**.** *Let $\mu^* = \max_{a\in\mathcal{C}} \mu_a$. For each arm $a \in \mathcal{C}$, let $\Delta_a = \mu^* - \mu_a$. The expected regret of UCB sampling algorithm is as follows:*

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{UCB}}(T)\right] = O\left(\sum_{a\in\mathcal{C}:\mu_a<\mu^*} \frac{\log T}{\Delta_a} + \Delta_a\right).$$

The expectation is over the randomness of the data.

3.1 Private UCB Sampling: Algorithm and Analysis

In Algorithm 1 we modify the UCB sampling algorithm to obtain an ϵ-differentially private variant. Notice that for each arm $a \in \mathcal{C}$ the average reward $r_a(t)$, is the only term that depends directly on the data set whose privacy we want to protect. So, if we can ensure that this sequence, $r_a(t)$, $t \in [T]$ is ϵ/k-differentially private for each arm a, then immediately we have ϵ-differential privacy for the complete algorithm. We invoke the tree based aggregation algorithm from Section 2.1 to make these sequences private. Additionally, to counter the noise added to the empirical mean, we loosen the confidence interval for the biases of each arm.

3.1.1 Privacy Analysis

Theorem 3 (Privacy guarantee)**.** *Algorithm 1 is ϵ-differentially private.*

Proof. The algorithm only accesses the reward for its computation via the tree based aggregation scheme (see Section 2). Since, there are k-arms, we maintain k separate trees, each of which is guaranteed to be $\epsilon_0 = \frac{\epsilon}{k}$ differentially private. Using the composition property of differential privacy, we immediately conclude that Algorithm 1 is ϵ-differentially private. □

3.1.2 Regret Analysis

The expected regret of the algorithm is given by $\mathbb{E}[\sum_{a\in\mathcal{C}:\mu_a<\mu_{a^*}} \Delta_a n_a(T)]$. Hence, if our algorithm limits the pulls of the bad arms, we are done. Our regret analysis proceeds as follows, first we bound the amount of noise that can be present in any of the total rewards $r_a(t)$. And later using this bound, we show that the number of times the suboptimal arms get pulled is small. We bifurcate the analysis of the each of the suboptimal arms into exploration and exploitation phase. We argue that in case of bad arms, after getting pulled for $O\left(\frac{k\log^2 T\log(kT)}{\epsilon\Delta_a^2}\right)$ rounds the arm is not selected again with high probability. The main arguments in this analysis follow the general sequence of arguments in the analysis for non-private UCB sampling. (See [9] for a comparison.)

Algorithm 1 Differentially Private UCB Sampling

Input: Time horizon: T, arms: $\mathcal{C} = \{a_1, \cdots, a_k\}$, privacy parameter: ϵ, failure probability: γ.

1: Create an empty tree Tree_{a_i} with T-leaves for each arm a_i. Set $\epsilon_0 \leftarrow \epsilon/k$.
2: **for** $t \leftarrow 1$ to k **do**
3: Pull arm a_t and observe reward $f_t(a_t)$.
4: Insert $f_t(a_t)$ into Tree_{a_t} via *tree based aggregation* (see Section 2.1) with privacy parameter ϵ_0.
5: **Number of pulls:** $n_{a_t} = 1$.
6: **end for**
7: Confidence relaxation:$\Gamma \leftarrow \frac{k\log^2 T\log((kT\log T)/\gamma)}{\epsilon}$.
8: **for** $t \leftarrow k+1$ to T **do**
9: **Total reward:** $r_a(t) \leftarrow$ Total reward computed using Tree_a, for all $a \in \mathcal{C}$.
10: Pull arm $a^* = \arg\max_{a\in\mathcal{C}} \left(\frac{r_a(t)}{n_a} + \sqrt{\frac{2\log t}{n_a}} + \frac{\Gamma}{n_a}\right)$ and observe $f_t(a^*)$.
11: **Number of pulls:** $n_{a^*} \leftarrow n_{a^*} + 1$.
12: Insert $f_t(a^*)$ into Tree_{a^*} using *tree based aggregation* and privacy parameter ϵ_0.
13: **end for**

Theorem 4 (Utility guarantee)**.** *Let $\{\mu_a : a \in \mathcal{C}\}$ be the biases of the k-arms in the set $\mathcal{C}$. Let $\mu^* = \max_{a\in\mathcal{C}} \mu_a$ and for each arm $a \in \mathcal{C}$, $\Delta_a = \mu^* - \mu_a$. With probability at least $1-\gamma$ (over the randomness of the algorithm), the expected regret (over the randomness of the data) is as follows:*

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{Priv-UCB}}(T)\right] = O\left(\sum_{a\in\mathcal{C}:\mu_a<\mu^*} \frac{k\log^2 T\log(kT/\gamma)}{\epsilon\Delta_a} + \Delta_a\right).$$

In the following lemma, we bound the error in the computation of the total reward in Line 9 in Algorithm 1, introduced due to privacy.

Lemma 5. *For all arms $a \in \mathcal{C}$ and all time step $t \in [T]$, w.p. $\geq 1-\gamma$ (over the randomness of the algorithm), the error in the computation of the total reward for a till time t is at most $\frac{k\log^2 T\log((kT\log T)/\gamma)}{\epsilon}$.*

The proof of this lemma is given in Appendix B.

Lemma 6. *For a given arm $a \in \mathcal{C}$, if the mean $\mu_a < \mu^*$ and $\Delta_a = \mu^* - \mu_a$, then w.p. $\geq 1-\gamma$ (over the randomness of the algorithm) $\mathbb{E}[n_a(T)] = O\left(\frac{\log^2 T \log(kT/\gamma)}{\epsilon \Delta_a^2}\right) + \zeta$. Here ζ is a fixed positive constant independent of the problem parameters. The expectation is over the randomness of the data and $O(.)$ only hides multiplicative constants.*

Proof. This lemma provides us with an upper limit on the expected number of pulls of the suboptimal arms. For each arm we partition the analysis into two phases (*exploration phase* and *exploitation phase*). For an arm a, such that $\mu_a < \mu^*$, we show that the exploration phase lasts for $s_a = \frac{8k \log^2 T \log((kT \log T)/\gamma)}{\epsilon \Delta_a^2}$ many pulls. Once exploration phase is over, the exploitation phase starts. If $n_a(T) \leq s_a$, then we are done, since it means the arm was never finished the exploration phase. Otherwise, we show that the probability of pulling an arm after its exploration phase is very low. In principle, what we show that for the bad arms, the expected number of pulls for those arms is bounded during its exploration phase.

Let $X_a(t)$ be the true total reward and $\mathsf{Noise}_a(t) = r_a(t) - X_a(t)$ for an arm a. For the ease of notation, let $\Gamma = \frac{k \log^2 T \log((kT \log T)/\gamma)}{\epsilon}$. At time step $t \in [T-1]$, an arm a is pulled over a^* if the following is true.

$$\begin{aligned}
&\frac{r_{a^*}(t)}{n_{a^*}(t)} + \sqrt{\frac{2\log t}{n_{a^*}(t)}} + \frac{\Gamma}{n_{a^*}(t)} \\
&\leq \frac{r_a(t)}{n_a(t)} + \sqrt{\frac{2\log t}{n_a(t)}} + \frac{\Gamma}{n_a(t)} \\
&\Leftrightarrow \frac{X_{a^*}}{n_{a^*}(t)} + \sqrt{\frac{2\log t}{n_{a^*}(t)}} + \frac{\Gamma}{n_{a^*}(t)} + \frac{\mathsf{Noise}_{a^*}(t)}{n_{a^*}(t)} \\
&\leq \frac{X_a}{n_a(t)} + \sqrt{\frac{2\log t}{n_a(t)}} + \frac{\Gamma}{n_a(t)} + \frac{\mathsf{Noise}_a(t)}{n_a(t)} \qquad (2)
\end{aligned}$$

It can be easily shown that (2) is true, only if at least one of the following equations hold.

$$\frac{X_{a^*}}{n_{a^*}(t)} \leq \mu^* - \sqrt{\frac{2\log t}{n_{a^*}(t)}} \qquad (3)$$

$$\frac{X_a}{n_a(t)} \geq \mu_a + \sqrt{\frac{2\log t}{n_a(t)}} \qquad (4)$$

$$\begin{aligned}
&\mu^* + \frac{\Gamma}{n_{a^*}(t)} + \frac{\mathsf{Noise}_{a^*}(t)}{n_{a^*}(t)} \\
&< \mu_a + \frac{\Gamma}{n_a(t)} + 2\sqrt{\frac{2\log t}{n_a(t)}} + \frac{\mathsf{Noise}_a(t)}{n_a(t)} \qquad (5)
\end{aligned}$$

Directly by the use of Chernoff bound, we can show that with probability at least $1 - 2t^{-4}$, (3) and (4) are false. To ensure that (5) does not hold, it suffices to ensure the following.

$$\begin{aligned}
&\frac{\Gamma}{n_a(t)} + 2\sqrt{\frac{2\log t}{n_a(t)}} + \frac{\mathsf{Noise}_a(t)}{n_a(t)} \leq \frac{\Gamma}{n_{a^*}(t)} + \frac{\mathsf{Noise}_{a^*}(t)}{n_{a^*}(t)} + \Delta_a \\
&\Leftrightarrow 2\sqrt{\frac{2\log t}{n_a(t)}} + \frac{\Gamma + \mathsf{Noise}_a(t)}{n_a(t)} - \left(\frac{\Gamma + \mathsf{Noise}_{a^*}(t)}{n_{a^*}(t)}\right) \leq \Delta_a \qquad (6)
\end{aligned}$$

From Lemma 5 we know that during the execution of the algorithm, w.p. $\geq 1-\gamma$, $|\mathsf{Noise}_{a^*}(t)|$ and $|\mathsf{Noise}_a(t)|$ are at most Γ. Therefore, to ensure (6) it suffices to ensure the following.

$$2\sqrt{\frac{2\log t}{n_a(t)}} + \frac{\Gamma + \mathsf{Noise}_a(t)}{n_a(t)} \leq \Delta_a \qquad (7)$$

If for some $0 < \nu < 1$ we have, $2\sqrt{\frac{2\log t}{n_a(t)}} \leq \nu\Delta_a$ and $\frac{\Gamma + \mathsf{Noise}_a(t)}{n_a(t)} \leq (1-\nu)\Delta_a$, then we can claim (7). Hence to ensure (7), by setting $y = \sqrt{\frac{\log(t)}{\Gamma}}$ (assume $T > 3$) it suffices to have $n_a(t) \geq \frac{8\Gamma}{\Delta_a^2}$ since $\Delta_a < 1$. Recall that this is the exactly the threshold for the exploration phase for the arm a. We now focus our attention to the term $\mathbb{E}[n_a(T)]$ we initially intended to bound. It is easy to see the following.

$$\begin{aligned}
\mathbb{E}[n_a(T)] &\leq \left\lceil \frac{8\Gamma}{\Delta_a^2} \right\rceil + \sum_{t=\frac{8\Gamma}{\Delta_a^2}}^{T} \Pr[\text{Pulling arm } a \text{ at time } t] \\
&\leq \left\lceil \frac{8\Gamma}{\Delta_a^2} \right\rceil + \sum_{t=\frac{8\Gamma}{\Delta_a^2}}^{T} \Pr[\exists n_a(t), n_{a^*}(t) \text{ s.t. (2) holds}] \\
&\leq \left\lceil \frac{8\Gamma}{\Delta_a^2} \right\rceil + \sum_{t=\frac{8\Gamma}{\Delta_a^2}}^{T} \sum_{n_{a^*}=1}^{t} \sum_{n_a=\frac{8\Gamma}{\Delta_a^2}}^{t} 2t^{-4} \\
&\leq \left\lceil \frac{8\Gamma}{\Delta_a^2} \right\rceil + \sum_{t=1}^{T} 2t^{-2} \leq \frac{8\Gamma}{\Delta_a^2} + \zeta \qquad (8)
\end{aligned}$$

This completes the proof Lemma 6. □

Proof of Theorem 4. In order to obtain the final regret guarantee of Theorem 4 and conclude the proof, we notice that, $\mathbb{E}\left[\mathsf{Regret}_{\mathsf{Priv-UCB}}(T)\right] = \mathbb{E}\left[\sum_{a \in \mathcal{C}: \mu_a < \mu_{a^*}} \Delta_a n_a(T)\right]$. Using Lemma 6 in the expression above concludes the proof. □

4 PRIVATE THOMPSON SAMPLING

In this section we study a different flavor of bandit learning algorithms called *Thompson sampling*. Historically,

Thompson sampling [25] is much older than UCB sampling style algorithms, dating back to early 20^{th} century. Although Thompson sampling is a very powerful heuristic and often outperforms UCB sampling in experimental setups, until recently little was known about its regret guarantees. [2] provided the first regret analysis for Thompson sampling and they showed that it has regret logarithmic in the time horizon T. We provide a differentially private variant of the Thompson sampling algorithm. One reason for studying private Thompson sampling along with private UCB sampling is because it is known that in the non-private world the experimental performance of Thompson sampling is much better than UCB sampling. Moreover it demonstrates properties like stability to delayed feedback. (See [8] for details.) The question we want to answer is whether we can obtain a differentially private variant of Thompson sampling which preserves the asymptotic regret guarantee of as the same order as the non-private algorithm and also demonstrate similar experimental properties.

Background on Thompson sampling. The basic Thompson sampling heuristic is extremely simple. Suppose there are k-arms $\mathcal{C} = \{a_1, \cdots, a_k\}$ which give Bernoulli rewards (i.e., $\{0,1\}$ rewards) and have biases $\mu_{a_1}, \cdots, \mu_{a_k}$. Let $r_{a_1}(t), \cdots, r_{a_k}(t)$ be the number of ones and $n_{a_1}(t), \cdots, n_{a_k}(t)$ be the total number of arm pulls, upto time t. The rule to pick the arm for each round is: Sample $\theta_i \sim \mathsf{Beta}(r_{a_i}(t)+1, n_{a_i}(t) - r_{a_i}(t) + 1)$ for $i \in [k]$ for $i \in [k]$ and pull the arm corresponding to the highest θ_i. Based on the result of the arm pull, the posterior distribution for that particular arm gets updated.[2] showed that the expected regret for this algorithm (over the randomness of the algorithm and the data) is $O\left(\sum_{a \in \mathcal{C}-a^*} \left(\frac{1}{\Delta_a^2}\right)^2 \log T\right)$, where a^* be the optimal arm and $\Delta_a = \mu_{a^*} - \mu_a$.

4.1 Private Thompson Sampling: Algorithm and Analysis

In this section we modify this generic Thompson sampling algorithm to obtain an ϵ-differentially private variant (Algorithm 2). We show that even with the privacy constraint our algorithm has almost the same regret as non-private algorithm with only poly $\log T$ overhead. This is the same setting in which [2] also proved their results. One interesting observation in the MAB problems is that all the sequential algorithms would encourage a downward bias, which means that if an arm does not give good results initially then it will not be pulled again, therefore it's empirical mean would be much lower than its true mean. UCB algorithm overcomes this problem by adding a monotonically decreasing function of the number of pulls ($n_a(t)$) to the empirical mean, thus in some sense balancing out this downward bias. But in case of Thompson sampling, we are randomizing our decision hence the bias correction mechanism is different, it is due to randomization. In Thompson sampling, the biggest challenge is to bound the number of mistakes in the initial rounds. Moreover to ensure differential privacy, we introduce additional randomness to the original Thompson sampling algorithm, it becomes even harder for us to analyze the number of mistakes in the initial rounds. So, we take a slightly different approach. We segregate the algorithm into *explicit exploration phase* and a *combined exploration and exploitation* phase. The idea in the exploration phase is to estimate the biases of the two arms (within sufficient confidence) without bothering about the number of mistakes made. In the joint exploration and exploitation phase, we use the standard Thompson sampling, except we make sure the algorithm only has differentially private access to the rewards. Similar to the private UCB algorithm (Algorithm 1), we use the *private tree based aggregation* (from Section 2.1) to ensure differential privacy. In the following section, for the convenience of notation we assume that $\mu^* = \mu_{a_1} > \mu_{a_2} \geq \cdots \geq \mu_{a_k}$.

4.1.1 Privacy Analysis

Theorem 7 (Privacy guarantee)**.** *Algorithm 2 is ϵ-differentially private.*

Proof. The proof of privacy is segregated into two parts. In the first part we argue that the *gap estimation* section is $\epsilon/2$-differentially private. In the second part we argue that the rest of the sections of the algorithm are combined $\epsilon/2$-differentially private. Using these two arguments and the composition property of differential privacy we argue that Algorithm 2 is ϵ-differentially private. Notice that in the first part, each arm i, $i \in [k]$ is pulled in batches of m-pulls, till the condition in Line 8 in Algorithm 2 is satisfied. If each of this batch is made $\frac{\epsilon}{2k}$-differentially private, then by *parallel composition* property of differential privacy, the first phase is $\epsilon/2$-differentially private. To guarantee $\frac{\epsilon}{2k}$-differential privacy for a given batch, we use Laplace mechanism from Section 2.1. The $\epsilon/2$-privacy guarantee for the second phase follows directly from the analysis of tree based aggregation scheme described in Section 2.1. □

4.1.2 Regret Analysis

In this section, we provide the regret analysis for our private Thompson sampling algorithm (Algorithm 2). The high-level structure of the analysis is as follows. First we establish that in the gap estimation phase, the estimated gap $\hat{\Delta}$ is within constant factor of the true gap $\Delta = \left|\mu_{a_{(1)}} - \mu_{a_{(2)}}\right|$ (i.e. the mean difference of the best arm and the second best arm). Moreover, we argue that the gap estimation runs for at most poly $\log T$ number of rounds. Second, assuming our estimation $\hat{\Delta}$ is reasonably accurate, we pull all the arms randomly for certain number of steps to ensure sufficient concentration of the empirical means around the true means. In the third and final stage, we analyze the noisy version of the classic Thompson sampling with *beta* prior. The analysis of this part resembles the analysis of [2]

Algorithm 2 Differentially Private Thompson Sampling

Input: Time horizon: T, arms: $\mathcal{C} = \{a_1, a_2, \cdots, a_k\}$, privacy parameter: ϵ.

Gap ($\Delta = |\mu_{a_1} - \mu_{a_2}|$) estimation
Initialize: Time counter: $\tau \leftarrow 0$, estimated gap: $\hat{\Delta} \leftarrow 1$, total pulls: $n_{a_1}, n_{a_2}, \cdots, n_{a_k}$.
repeat
  Pull each arm a_i $m = \frac{192k \log T}{\epsilon \hat{\Delta}^2}$ times to obtain average rewards $\tilde{\mu}_{a_i}$ $\forall a_i \in \mathcal{C}$. Increment $n_{a_i} \leftarrow n_{a_i} + m$ $\forall a_i \in \mathcal{C}$.
  Differentially private means: $\hat{\mu}_{a_i} \leftarrow \tilde{\mu}_{a_i} + \mathsf{Lap}\left(\frac{2k}{\epsilon m}\right), \forall a_i \in \mathcal{C}$.
  Set $\hat{\Delta} \leftarrow \hat{\Delta}/2$ and $\tau \leftarrow \tau + km$.
  Find the best and second best arms as, $a(1) = \underset{a_i \in \mathcal{C}}{\operatorname{argmax}} \mu_{a_i}$ and $a(2) = \underset{a_i \in \mathcal{C}, a_i \neq a_{(1)}}{\operatorname{argmax}} \mu_{a_i}$
until $|\hat{\mu}_{a(1)} - \hat{\mu}_{a(2)}| > \hat{\Delta}$
Create empty trees Tree_{a_i} with $(T-\tau)$-leaves $\forall a_i \in \mathcal{C}$. Set $\epsilon_0 \leftarrow \epsilon/2k$.
Random pullings of arms to build confidence
Confidence parameter: $\Gamma \leftarrow \frac{192 \log^3 T}{\epsilon}$.
Pull each arm a_i, $\frac{\Gamma}{\hat{\Delta}^2}$ times. Record the total rewards r_i $\forall a_i \in \mathcal{C}$. Insert r_i in Tree_{a_i} with privacy parameter ϵ_0, $\forall a_i \in \mathcal{C}$. Increment $n_{a_i} \leftarrow n_{a_i} + \frac{\Gamma}{\hat{\Delta}^2}$ and $\tau \leftarrow \tau + \frac{k\Gamma}{\hat{\Delta}^2}$.
Combined explore-exploit phase of Thompson sampling
for $t \leftarrow \tau + 1$ to T **do**
  Total reward: $r_a(t) \leftarrow$ Total reward computed using $\mathsf{Tree}_a, \forall a_i \in \mathcal{C}$. $\theta_a(t) \sim \mathsf{Beta}(r_a(t)+1, n_a(t) - r_a(t)+1), \forall a_i \in \mathcal{C}$. Pull arm $a^* = \arg\max_{a_i \in \mathcal{C}}(\theta_a(t))$ and observe reward $f_t(a^*)$, Insert $f_t(a^*)$ into Tree_{a^*} using *tree based aggregation* and privacy parameter ϵ_0,
  Number of pulls: $n_{a^*}(t) \leftarrow n_{a^*}(t) + 1$.
end for

closely. The overall regret guarantee is given in Theorem 8 below.

Theorem 8 (Utility guarantee)**.** *Let $\mu_{a_1}, \mu_{a_2}, \cdots, \mu_{a_k}$ be the biases of the k arms. Let $\Delta = \left|\mu_{a_{(1)}} - \mu_{a_{(2)}}\right|$, where $a(1) = \arg\max_{a_i \in \mathcal{C}} \mu_{a_i}$ and $a(2) = \arg\max_{a_i \in \mathcal{C}, a_i \neq a_{(1)}} \mu_{a_i}$. Then, for $T \geq 8 \max\{\log_2 \frac{1}{\Delta}, 1\}$ and $\epsilon < 1$ expected regret of Algorithm 2 (over the randomness of the data and the algorithm) is as follows:*

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{Priv-Thomspon}}(T)\right] = O\left(k \frac{\log^3 T}{\Delta^2 \epsilon^2}\right)$$

Let N_1, N_2 and N_3 denote the number of times the wrong arm is pulled in the *gap estimation* phase, *random pullings* phase and *combined explore and exploit* phase of Algorithm 2 respectively. We would bound the expected values of N_1, N_2 and N_3 using the following lemmas which would allow us to prove Theorem 8.

Lemma 9 (Bound on gap estimation phase)**.** *Following the notation of Theorem 8, then with probability at least $1 - 1/T^4$ (over the randomness of the algorithm and the data distribution), for $T \geq 8k \max\{\log_2 \frac{1}{\Delta}, 1\}$ and $\epsilon < 1$ the estimated gap $\hat{\Delta}$ in Algorithm 2 is in $[\Delta/3, 2\Delta]$, the expected number of times incorrect arm is pulled is, $\mathbb{E}[N_1] = O\left(\frac{k \log_2 T}{\epsilon}\right)$.*

The proof of this Lemma is provided in Appendix C.

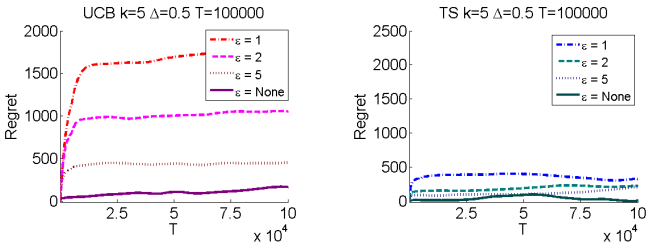
Lemma 10 (Bound on geometric random variables (Lemma 3 [2]))**.** *Let s_j be a random variable, index by $j \in \mathbb{Z}^+$ s.t. for fixed parameters $T \in \mathbb{Z}^+$ and $\mu \in [0, 1]$, with probability at least $1 - T^{-2}$, $\frac{s_j}{j} > \frac{\mu + y}{2}$. Let $y \in [0, \mu)$ be a predefined threshold and let $X(s_j, y)$ be the random variable that counts the number samples $w \sim \mathsf{Beta}(s_j + 1, j - (s_j + 1))$ that need to be drawn i.i.d. before w exceeds y. Using T as the fixed upper bound on $X(s_j, y)$, if $j \geq \frac{4 \log_2 T}{(\mu - y)^2}$, then $\mathbb{E}_s\left[\mathbb{E}_w\left[\min\{X(s_j, y), T\}\right]\right] = O\left(\frac{1}{T}\right)$.*

Lemma 11 (Bound on Beta random variable [Lemma 7 [2]])**.** *] Let us define event $E(t)$ as, $E(t) : \theta_a(t) \in [\mu_a - \frac{\Delta_a}{2}, \mu_a + \frac{\Delta_a}{2}], \forall a \in \mathcal{C}, a \neq a_1$. Then $\Pr(E(t), n_a(t) \geq \frac{32 \log_2 T}{\Delta^2}) \geq 1 - \frac{4(k-1)}{T^3}$, $\forall t$.*

Lemma 12 (Regret bound in the combined explore and exploit phase)**.** *Suppose N_3 is a random variable which counts the number of times the suboptimal arms are pulled in Algorithm 2. Then, over the randomness of the algorithm and the data, $\mathbb{E}[N_3] = O(1)$.*

Proof. In order to calculate the regret we count the number of times the sub-optimal arms get pulled since it upper bounds the regret. The main idea behind the proof of this lemma is that via random pulling of the arms in Algorithm 2, the arm a_1 is well separated from the other sub-optimal arms. Hence, in the combined explore and exploit phase, with high probability the optimal arm gets pulled almost always. We will use the proof technique from [2] for analyzing the number of pulls of the suboptimal arms.

We denote the set of suboptimal arms by $\mathbb{S}$ and $\mathbb{S} = a_2, \cdots, a_k$. We count the number of pulling of sub-optimal arm a_s, by the following scheme: count the number of pulls of arm a_s in between two successive pulls of arm a_1. First, recall that with probability at least $1 - T^{-4}$ (from Lemma 9), $\hat{\Delta}$ (the estimated gap) is within $[\Delta/3, 2\Delta]$. By the property of the tree based aggregation discussed earlier, the difference between the true total reward at time $t(j)$ and $r_a(t(j))$ is at most $\frac{\log_2^3 T}{\epsilon}$, with probability $1 - \text{negl}(T)$. Since by the beginning of combined explore and exploit phase, we have pulled arm a_1 at least $\frac{\Gamma}{\hat{\Delta}^2}$-times, it follows that with probability at least $1 - 2T^{-4}$, difference between the true total reward at time $t(j)$ and $r_{a_1}(t(j))$ is at most $\frac{\Delta}{4}$. Using Lemma 11 and the fact that we pulled all the arms for $\Gamma/\Delta^2 > \frac{32 \log_2 T}{\Delta_a^2}$, we observe that with probability atleast



(a) (b)

Figure 1: Simulation results for our differentially private algorithms UCB sampling (Algorithm 1) and Thompson sampling (Algorithm 2) for the number of arms $k = 5$ and $\Delta = 0.5$. Smaller ϵ indicates more privacy.

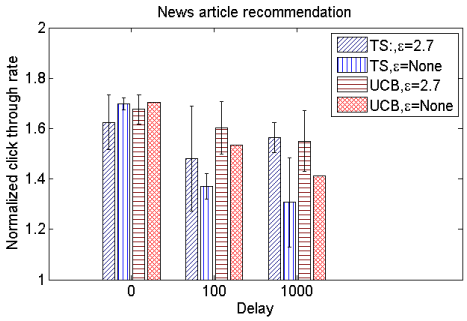


Figure 2: Comparison of different differentially private and non-private multi-armed bandit algorithms on Yahoo! front page News article recommender system. The click-through rates for each algorithm is normalized with respect a random algorithm.

$1-\frac{4k-2}{T^3}$, $\theta_a < \mu_a + \Delta_a/2$. Hence bundled up failure probabilities obtained so far for $\theta_{a_1} < \theta_{a_s}$ using union bound is $1-\frac{4}{T^2}$. Let j and $j+1$ be any two successive pulls of arm a_1, let $t(j)$ and $t(j+1)$ be the respective time epochs and denote $\Delta_s = \mu_{a_s} - \mu_{a_1}$.The Beta distribution for the sample $\theta_a$from arm a in between these pulls (not including the $(j+1)$-th pull is $\mathsf{Beta}(r_a(t(j))+1, n_a(t(j))-r_a(t(j))+1)$. Now a suboptimal arm a_s is pulled during round j and $j+1$ pull of arm a_1 if $\theta_{a_1} < \mu_{a_1} - \min_{a_s \in \mathbb{S}} \Delta_{a_s}$. In order to use Lemma 10 above, we set the threshold $y = \mu_{a_2} + \frac{\Delta}{2}$. Moreover, if we set $s_j = r_{a_1}(t(j))$, then we know that with probability at least $1 - 4T^{-2}$, $\frac{s_j}{j} > \mu_{a_1} - \frac{\Delta}{4}$. Hence, the conditions of Lemma 10 holds and summing over all T rounds, we complete the proof. □

Proof of Theorem 8. The expected regret of multi-armed bandit algorithms can be upper bounded by the number of times the suboptimal arm is pulled. We defined N_1, N_2 and N_3 to denote the number of times the wrong arm is pulled in the *gap estimation* phase ,*random pullings* phase of Algorithm 2 and *combined explore and exploit* phase of Algorithm 2 respectively. Thus we have,

$$\mathbb{E}\left[\mathsf{Regret}_{\mathsf{Priv-Thomspon}}(T)\right] = \mathbb{E}\left[N_1\right] + \mathbb{E}\left[N_2\right] + \mathbb{E}\left[N_3\right]$$

From Lemma 9 we know that $\mathbb{E}\left[N_1\right] = O\left(k\frac{\log T}{\epsilon^2}\right)$. From Lemma 9 and Algorithm 2, we know that $\mathbb{E}\left[N_2\right] = O\left(k\frac{\Gamma}{\Delta^2}\right)$, where $\Gamma = O\left(\frac{\log^3 T}{\epsilon}\right)$. Finally from Lemma 12, we know that $\mathbb{E}\left[N_3\right] = O\left(1\right)$. Combining these bounds, we get the bound on the expected regret. □

5 EXPERIMENTAL EVALUATION

In this section, we support the theoretical regret bounds for our algorithms (Algorithm 1 and 2) with empirical results. The experimental results show that there is a smooth trade-off between privacy and accuracy. As we increase our privacy parameter ϵ, the regret improves. We perform the simulation experiments on for stochastic multi-arm bandits, with rewards in $\{0, 1\}$. The k-arm private UCB sampling algorithm is described in Section 3. The 2-arm private Thompson sampling and its extension to k-arm private Thompson sampling are given in Section 4.1. The true underlying distribution of the arms are chosen as follows. The mean for the best arm is 0.9 and the other arms have biases of $0.9 - \Delta$ each, where $\Delta = 0.5$. We tune the parameters of our private algorithms, the *confidence parameter* (see Line 9 in Algorithm 1) for UCB and the number of *random pullings* (see Line 12 in Algorithm 1) for Thompson to enhance accuracy. Note, the parameter tuning does not violate privacy, since the access to the aggregated rewards is still based the on tree based aggregation scheme. All the UCB type sampling algorithms have a common parameter, the confidence interval; same as Line 9 in Algorithm 1. For our experiment on private UCB sampling (Algorithm 1), we use a particular confidence interval, given in [8], which seems to perform the best. The confidence interval is given as, $\frac{\sqrt{r_a(t)\log t}}{n_a(t)} + \frac{\log t}{n_a(t)}$, where $r_a(t)$ and $n_a(t)$ are the reward and number of pulls for arm $a \in \mathcal{C}$ up to time t respectively. For our experiments on private Thompson sampling (see Algorithm 2) we do not implement the *gap estimation* phase and for the second phase that involves *random pullings* (Line 10 of 2), use a smaller value for m.

Conclusions drawn from simulations. We observe in the plots that the regret for the private algorithms saturates after certain time, similar to that of their non-private counterparts (see Figure 1). Also notice that in the simulations, Thompson sampling tends to perform much better than UCB sampling.

5.1 Yahoo! Front Page Data set

In this section, we describe our results on *Yahoo! front page news article recommendation* data set. The data set contains 45,811,883 user visits to the *Today module* during first 10 days in May 2009. Each user click on a news article shown corresponds to a reward of one for that article. This data set has also been used by [21], [11] for bandit experiments. One property of this data set is that the displayed article is chosen uniformly at random from the candidate article pool allowing us to use an *unbiased offline* evalua-

tion method [21, 22]. The pool of articles is small (around 20 articles), but it is dynamic which means that the articles may be added or removed from this pool. For each visit, both the user and each of the candidate articles are associated with a feature vector of dimension 6. The feature vector acts as a context for the news article recommender and based on this context the most suited article can be chosen using a bandit algorithm. This is the contextual bandit setting. In this setting, in each of T rounds, a learner is presented with the context vector: $z_a \in \mathbb{R}^d$ for each arm $a \in \mathcal{C}$ and based on his previous observations and this new context vector, the learner needs to select one out of k actions. The learner's aim is to learn the relation between the reward and the context vector in an online fashion.

Differentially private contextual sampling (Algorithm 3 and Algorithm 4 in Appendix D.2 and D.3.) The private contextual UCB algorithm is adapted from the *LinUCB* algorithm in [21] and is similar to the basic UCB sampling algorithm, as it computes the expected reward of each arm and then chooses the arm with the highest upper confidence bound. The expected reward is given as $z_t^T \theta_t$, where θ_t is estimated using ridge regression and the confidence bound is given as $\sqrt{z_a(t)^T A_t z_a(t)}$, which is the Mahalanobis distance of the context vector with covariance matrix A. On the other hand, the private contextual Thompson sampling algorithm is a differentially private version for the algorithm provided by [3]. In regular Thompson sampling, for each round we choose an arm according to its posterior probability of having the best parameter. A natural generalization of Thompson Sampling for contextual bandits is to use Gaussian prior and Gaussian likelihood function. The extension of these algorithms to private algorithms is straightforward. In both the private algorithms, we restrict our access to the parameters which aggregate over each time stamp and use *tree based aggregation* scheme to retrieve those parameters. We give the details of these algorithms in Appendix D.2 and D.3.

Conclusions on experiments with Yahoo! front page data set. The results for this experiment are summarized in Figure 2. We find that the private algorithms do not perform much worse than the non-private algorithms. Since, the feature vector is of length 6 by setting the privacy ϵ_0 of each parameter as 0.1, the total privacy measured in terms of the total privacy parameter $\epsilon = 2.7$. We also investigate the performance of the algorithms with respect to delays. We have considered the delay values in $\{0, 100, 1000\}$. When the input data does not have any delay in the feedback, the private algorithms perform slightly worse than the non-private counter parts and as the delay increases the performance of the non-private algorithms is hurt more than the private algorithms.

References

[1] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40, 2010.

[2] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, 2012.

[3] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. *arXiv preprint arXiv:1209.3352*, pages 1–29, 2012.

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

[5] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012.

[6] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like: " privacy risks of collaborative filtering. In *IEEE Symposium on Security and Privacy*, 2011.

[7] TH Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. In *ICALP*. 2010.

[8] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

[9] Kamalika Chaudhuri. Topics in online learning: Lecture notes. 2011.

[10] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*. MIT Press, 2008.

[11] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. A case study of behavior-driven conjoint analysis on yahoo!: front page today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104. ACM, 2009.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[13] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *STOC*, 2010.

[14] Cynthia Dwork, Moni Naor, Omer Reingold, Guy Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[15] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[16] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. *arXiv preprint arXiv:1109.0105*, 2011.

[17] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24.1–24.34, 2012.

[18] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[19] Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *International Conference on Data Mining Workshops*, 2010.

[20] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[21] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[22] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

[23] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[24] Adam Smith and Abhradeep Thakurta. Nearly optimal algorithms for private online learning in full-information and bandit settings. In *NIPS (To appear)*, 2013.

[25] 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.