Private Stochastic Multi-arm Bandits: From Theory to Practice

Nikita Mishra NMISHRA@CS.UCHICAGO.EDU
University of Chicago

Abhradeep Thakurta B-ABHRAG@MICROSOFT.COM
Stanford University and Microsoft Research

Abstract

In this paper 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 (Dwork et al., 2010; Jain et al., 2012; Smith and Thakurta, 2013). 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). Moreover, through our experiments on simulated and real-world data sets, we empirically show the effectiveness of our algorithms.

Note: A version of this paper with the technical details appears in the appendix.

1. Introduction

Consider a set of k arms $\mathcal{C} = \{a_1, \cdots, a_k\}$. At each time step $t \in [T]$ an arm $a \in \mathcal{C}$ gets pulled, where T is the time horizon. Corresponding to the pulled arm a, a reward of $f_t(a) \in \mathbb{R}$ is awarded by the environment. The objective is to design an algorithm (or learner) which maximizes the total reward (i.e., $\sum_t f_t(a)$) over all time steps T. The *only* information the algorithm gets while interacting with the system is the set of rewards for the arms it has pulled during the period $1, \cdots, T$. This class of online learning algorithms which work under *partial feedback* (i.e., at any time step t, it cannot see the rewards for any of the arms it has not pulled) are called bandit algorithms.

Bandit algorithms have been broadly categorized into two classes: i) Adversarial bandits (where the environment chooses the rewards adversarially), and ii) Stochastic bandits (where the environment chooses the rewards from an unknown but fixed distribution). Both stochastic bandits and adversarial bandits have been studied extensively in the online learning literature. (See (Shalev-Shwartz, 2011) or (Bubeck and Cesa-Bianchi, 2012) for a detailed survey.)

Recently, (Smith and Thakurta, 2013) analyzed bandit algorithms under the constraint of *differential privacy* and provided nearly optimal error guarantees for adversarial bandits. Informally speaking, differential privacy ensures that from the output of the algorithm (i.e., the arms the algorithm pulls), an adversary's[1] information gain about the rewards assigned to $\mathcal{C}$ by the environment at any time step t is small. In this work, we study the problem of stochastic bandits while preserving differential privacy. We provide nearly optimal algorithms for differentially private stochastic bandits, and provide experimental evidence for the effectiveness of our algorithms.

Often privacy forms a serious bottleneck in the usage of bandit learning algorithms in practice. As a concrete example, consider the scenario of recommendation (or advertisement) system. One can view the set of candidate recommendations for the user, as the set of arms $\mathcal{C}$. If the user clicks on one of the recommendations, then a particular reward is given back to the recommendation system based on user's click. Since user preferences can be potentially sensitive, a recommendation system can leak a lot of potentially sensitive information about a user. Attacks on real recommendation systems (Calandrino et al., 2011) have heightened the privacy concerns to a large extent. Our bandit algorithms provide provable privacy guarantees to the individuals in the data set whose data are used to train recommendation systems.

[1] Here adversary is the one who wants to extract information about rewards from the environment. N.B. This should not be confused with adversarial bandit learning, where the environment chooses the rewards so as to minimize the total reward for the bandit algorithm.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

In this work we extend the ideas from (Smith and Thakurta, 2013; Dwork et al., 2010; Jain et al., 2012) for private online learning under full-information and adversarial bandits to the case of stochastic bandits. The two main novelties of our results are: i) We show strong experimental evidence that our private algorithms are useful on real-scale data, and ii) We extend our algorithms to the general case of contextual bandits. Contextual bandit is a generalization of the basic stochastic bandit formulation above, where at each time t, a context vector $z_a(t)$ is provided for each arm $a \in \mathcal{C}$. The reward for an arm a at time step t has a distribution parameterized by $z_a(t)$. (For details see Section 4.) The underlying algorithm we use in our work is the *upper confidence bound* (UCB) sampling algorithm, initially proposed by (Auer et al., 2002). Both for the contextual and the context free case of the UCB sampling, the privacy analysis is similar to that of (Smith and Thakurta, 2013). However, we need to provide a fresh analysis as a direct black box reduction is not possible.

Algorithmic idea. Stochastic multi-armed bandit algorithms usually run in 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 distributions from which the reward for each arm is drawn. In the second phase it uses this information to guide the choice of arms in the later rounds. In order to ensure differential privacy, we are required to introduce some randomness in the observed rewards, but a direct noise addition will grossly corrupt the estimates for the arms. We address this issue by increasing the number of rounds used by the algorithm to estimate the means. The exact details are slightly more complicated and are discussed in Section 3.

Privacy semantics. We now focus on the semantics of differential privacy in the setting where the data points (the rewards) arrive online in a stream at every time step. This setting was first studied by (Dwork et al., 2010) and then followed by (Jain et al., 2012) and (Smith and Thakurta, 2013). 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. (Jain et al., 2011) 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. (Smith and Thakurta, 2013) 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 (Shalev-Shwartz, 2011)). In contrast, stochastic bandit algorithms enjoy a regret of $O(\log T)$. In this work we obtain the *first* and *nearly optimal* regret guarantees by building on the algorithmic technique of (Smith and Thakurta, 2013) to the case of stochastic bandit problems. Since, stochastic bandit algorithms have a very different flavor than adversarial bandit algorithms, our results do not follow directly from (Smith and Thakurta, 2013).

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 follow 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, we provide experimental evidence to corroborate our theoretical guarantees.

1.1. Our Contributions

Here, we provide an overview of our contributions.

- **Differentially private UCB sampling:** We provide a differentially private variant of UCB sampling algorithm 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 (Dwork et al., 2009; Chan et al., 2010). Our utility analysis goes via carefully analyzing the *exploration phase* of the algorithm, where it estimates the means of the arms. As a consequence, we provide a version of UCB sampling algorithm that is robust to noise.
- **Differentially private contextual bandits:** We provide the first differentially private algorithm for contextual bandits. We modify our basic private UCB algorithm to the contextual case, and use the algorithm of (Li et al., 2010) as the basic building block. Although we do not provide any formal utility analysis, we show the effectiveness of our private algorithm on real-world data sets.
- **Experimental evaluation:** We provide a thorough experimental evaluation of the private UCB sampling on both simulated and real world data sets (Yahoo! news recommendation data). On the simulate data sets we show that our basic private UCB sampling algorithm perform as good as the non-private counter part. For

the contextual UCB algorithm, we show that on the Yahoo! news recommendation data set, our algorithm perform comparable to the non-private counterpart of (Li et al., 2010).

2. Background and Problem Definition

2.1. Background on Differential privacy

In this section we provide a short overview of differential privacy. Let $\mathcal{D} = \langle f_1, \cdots, f_T \rangle$ 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 by Algorithm $\mathcal{A}$.

Definition 1 (Differential privacy (Dwork et al., 2006))**.** *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 for the definition is that the guarantee should hold for every pair of neighboring data sets. Therefore, for the regret analysis of our algorithm $\mathcal{A}$ although we can assume that the rewards come from some underlying distribution, but we *cannot* make any stochastic assumption on the reward functions for privacy guarantee. Next, we discuss some of the basic tools for designing differentially private algorithms.

Laplace mechanism and Gamma mechanism. Laplace mechanism (Dwork et al., 2006) and Gamma mechanism (Chaudhuri and Monteleoni, 2008; Chaudhuri et al., 2011) are simple 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. (See (Dwork et al., 2006) for the proof.)

Gamma mechanism is also very similar to Laplace mechanism. The only difference being that we now need to work with a vector valued function $f : \mathcal{U}^* \to \mathbb{R}^p$. Analogous to the 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 (Chaudhuri et al., 2011) for the proof.)

Tree based aggregation. Initially proposed by (Dwork et al., 2009; Chan et al., 2010), this aggregation scheme is extremely effective in releasing private continual statistics over a data stream. Suppose at every time step $t \in [T]$, one entry from the 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 the full version.) 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 $t \in [T]$.

2.2. Background on Stochastic Multi-arm Bandits and Problem Definition

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 cost 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 $\max_{x \in \mathcal{C}} \sum_{t=1}^{T} f_t(x)$. (For a detailed discussion, see (Shalev-Shwartz, 2011).) A natural measure of the utility of $\mathcal{A}$ is *regret*, defined as:

$$\mathsf{Regret}_{\mathcal{A}}(T) = \max_{x \in \mathcal{C}} \sum_{t=1}^{T} f_t(x) - \sum_{t=1}^{T} f_t(x_t).$$

Under this umbrella of regret minimization, 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 observes much lesser information from the environment and only gets to see the evaluation of f_t at x_t.

Problem Statement. Let f_t be defined as $f_t : \mathcal{C} \to [0, 1]$ for all $t \in [T]$, 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 polylogarithmically on 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 Upper Confidence Bound Sampling

Upper Confidence Bound (UCB) sampling by [(Auer et al., 2002)] is an algorithm for stochastic multi-arm bandit (MAB) problems, which despite being very simple gives very strong utility guarantees. The regret for UCB $O^*(\log T)$ infact matches the asymptotic lower given by (Lai and Robbins, 1985) 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 (Agarwal et al., 2010; Flaxman et al., 2005)). Recently (Smith and Thakurta, 2013) provided differentially private algorithms for adversarial bandit problems, which are almost optimal in the parameter T. In this section, for stochastic MAB, we provide a differentially private UCB algorithm whose expected regret is polylogarithmic in T. Before we move to the private UCB algorithm, we provide a brief overview of the non-private version.

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.,

$$\arg\max_{a \in \mathcal{C}} \frac{r_a(t)}{n_a(t)} + \sqrt{\frac{2 \log t}{n_a(t)}}. \quad (2)$$

Theorem 2 (Regret for non-private UCB Sampling [(Auer et al., 2002))**.** *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 means of each arm.

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

We defer the proof of the privacy guarantee to the full version.

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 split the analysis of the each of the suboptimal arms into the exploration and the exploitation phase. We argue that in case of any bad arm, 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 (Chaudhuri, 2011) for a comparison.) Although our algorithm assumes that we know the time horizon T, it can be easily extended to unknown horizon using the standard doubling trick. Thus, we obtain the following utility guarantee.

Theorem 4 (Utility guarantee)**.** *Let $\{\mu_a : a \in \mathcal{C}\}$ be the means 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:*

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

4. Private Contextual Bandits and Linear UCB

In the contextual setting, at each time step t, the learner receives a context vector $z_a(t)$ for each arm $a \in \mathcal{C}$. For a given arm $\hat{a}$ pulled by the algorithm, the expectation of the reward $f_t(\hat{a})$ (given the context vector $z_{\hat{a}}(t)$) equals $\mathbb{E}[f_t(\hat{a})|z_{\hat{a}}(t)] = \langle z_{\hat{a}}(t), \theta^*_{\hat{a}}\rangle$. Here θ^*_a is a hidden parameter vector corresponding to each arm $a \in \mathcal{C}$.

The private contextual UCB algorithm is adapted from the *LinUCB* algorithm in (Li et al., 2010) 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 for each arm a is computed by $\langle z_a(t), \theta_a(t)\rangle$, where $\theta_a(t)$ is estimated using ridge regression and the confidence bound is estimated by $\sqrt{z_a(t)^T A_t z_a(t)}$, which is the Mahalanobis distance of the context vector with covariance matrix $A_t = \sum_{\tau=1}^{t} z_a(\tau)z_a(\tau)^T$. The extension of this algorithm to private variant is fairly immediate. The idea is to restrict our access to the parameters which aggregate over the time steps, and use *tree based aggregation* scheme to retrieve those parameters while preserving differential privacy. The details of the algorithm is given in Algorithm 2. An interesting direction for future work is to give have powerful theoretical guarantee for the private contextual bandit.

Algorithm 2 Private Contextual UCB Sampling

Input: Time horizon: T, arms: $\mathcal{C} = \{a_1, \cdots, a_k\}$, privacy parameter: ϵ, expore/exploit parameter: α, Context vector length: d.

Initialize: $A = \mathbb{I}_d$ (*Identity matrix of size-d*), $\mu = 0_d$ (*Vector of length-d with all 0 entries*), $b = 0_d$.
Create empty trees $\mathsf{Tree}_{A_{i,j}} \forall i \le j \le d$ and $\mathsf{Tree}_{b_i} \forall i \le d$ with (T)-leaves. Set $\epsilon_0 \leftarrow \frac{2\epsilon}{(d^2+3d)}$.
for $t \leftarrow 1$ to T **do**
  Receive **Arm context:** $z_a(t) \forall a \in \mathcal{C}$.
  Receive $\tilde{A_{i,j}} \leftarrow$ from $\mathsf{Tree}_{A_{i,j}}$, set $\tilde{A}_{i,j} = \tilde{A}_{j,i}$ and Receive $\tilde{b_i} \leftarrow$ from Tree_{b_i}.
  if $\tilde{A}$ is positive definite **then**
    Pull arm $a^* = \arg\max_{a\in\mathcal{C}}(z_a(t)^T\tilde{A}^{-1}\tilde{b} + \alpha\sqrt{z_a(t)^T\tilde{A}z_a(t)})$, observe reward $f_t(a^*)$.
  else
    Pull arm $a^* = \arg\max_{a\in\mathcal{C}}(z_a(t)^T\tilde{b} + \alpha\sqrt{z_a(t)^T z_a(t)})$, observe reward $f_t(a^*)$.
  end if
  Insert $z_{a^*}(i)z_{a^*}(j)$ into $\mathsf{Tree}_{A_{i,j}}$ $\forall i \le j \le d$ and
  $z_{a^*}(i)f_t(a^*)$ into Tree_{b_i} $\forall i \le d$, using *tree based aggregation* and privacy parameter ϵ_0.
end for

5. Experimental Evaluation

In this section, we support the theoretical regret bounds for our algorithms (Algorithms 1 and 2) with empirical results, first on a simulated data and then on a real world data (*Yahoo! webscope front page news article recommendation*). The experimental results show that there is a smooth tradeoff between privacy and accuracy. As we increase our privacy parameter ϵ, the regret improves. We also perform experiments to investigate the effect of delayed feedback. In this context, delayed feedback means that the parameters are updated after some time lag, rather than immediately after each observation. We observe that our private algorithms are reasonably stable w.r.t. delayed feedback (see Section 5.2).

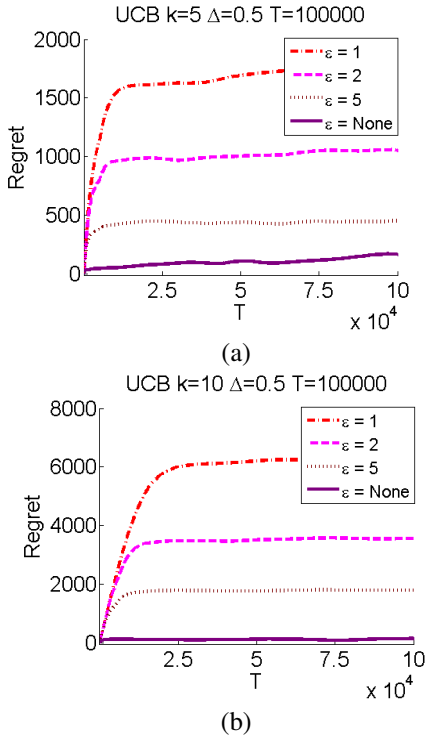

Figure 1. Results for our differentially private algorithms UCB sampling (Algorithm 1) with number of arms $k \in \{5, 10\}$ and $\Delta = 0.5$.

5.1. Experiments on Simulated Data set

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. We perform the experiments for $k \in \{5, 10\}$. The true underlying distribution of the arms are chosen as follows. The bias for the best arm is 0.9 and the other arms have biases of $0.9 - \Delta$ each, where $\Delta = 0.5$.

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). Similar to the non-private counter parts, the error is accumulated mainly in the exploration phase. Once the exploration phase is over the regret remains fairly stable.

5.2. 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 (Li et al., 2010), (Chu et al., 2009) for bandit experiments. One property

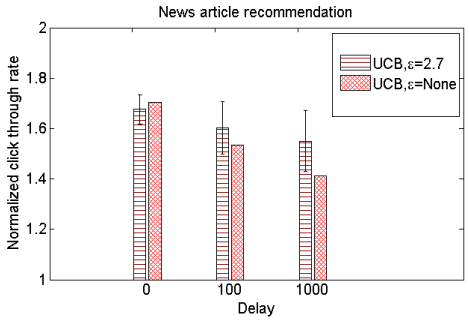

Figure 2. Comparison of different differentially private and non-private multi-armed bandit algorithms (Algorithm 2) on Yahoo! front page News article recommender system. The click-through rates for each epsilon is normalized with respect a random algorithm. The delay is with respect to the number of rows skipped before updating parameters.

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* evaluation method (Li et al., 2010; 2011). 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 six. 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.

Conclusions on experiments with Yahoo! front page data set. The results for this experiment are summarized in Figure 2. We find that the private algorithm does not perform much worse than the non-private algorithm. We set $\epsilon_0 = 0.1$ and since $d = 6$, we obtain that the privacy parameter $\epsilon = 2.7$. (See Algorithm 2) We also investigate the performance of the algorithm with respect to delays. It basically means that for a delay of $\tau \in \mathbb{N}$ steps, the parameter update at time t considers previous $t - (t \mod (\tau + 1) + 1)$ observations. We have considered the delay values in $\{0, 100, 1000\}$. When the input data does not have any delay in the feedback, the private algorithm perform slightly worse than the non-private counter parts and as the delay increases the performance of the non-private algorithm is hurt more than the private algorithms.

References

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

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

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

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.

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

Kamalika Chaudhuri. Topics in online learning: Lecture notes. 2011.

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

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *JMLR*, 12:1069–1109, 2011.

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.

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.

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.

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

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.

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

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

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

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.

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.

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

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