Local Differential Privacy for Sequential Decision Making in a Changing Environment

Pratik Gajane

Eindhoven University of Technology
pratik.gajane@gmail.com

Abstract

We study the problem of preserving privacy while still providing high utility in sequential decision making scenarios in a changing environment. We consider abruptly changing environment: the environment remains constant during periods and it changes at unknown time instants. To formulate this problem, we propose a variant of multi-armed bandits called non-stationary stochastic corrupt bandits. We construct an algorithm called SW-KLUCB-CF and prove an upper bound on its utility using the performance measure of *regret*. The proven regret upper bound for SW-KLUCB-CF is near-optimal in the number of time steps and matches the best known bound for analogous problems in terms of the number of time steps and the number of changes. Moreover, we present a provably optimal mechanism which can guarantee the desired level of local differential privacy while providing high utility.

Introduction

Several practically relevant applications including recommender systems, Internet advertising have been formulated as sequential decision making problems using the framework of multi-armed bandits. The importance of privacy in such sequential decision making problems has been extensively discussed in the literature (see for example, **???**).

Differential privacy, introduced by **?**, is one of the popular approaches to address such privacy concerns. In sequential decision making problems, algorithms providing differential privacy preserve data privacy by adding appropriate statistical noise to the data. **?** extend this notion to *local differential privacy* in which data remains private even from the algorithm. The main difference between global and local differential privacy is whether privacy is to be maintained from the algorithm or the (possibly unintended) recipient of the output of the algorithm. In global differential privacy, noise is added by the algorithm so the output does not reveal private information about the input. In local differential privacy, noise is added to the input of the algorithm so that privacy is maintained even from the algorithm.

To understand the motivation for local differential privacy, let us consider the practical application of Internet advertising [1]. An advertising system receives, as input, feedback from the users which may reveal private information about them. The advertising system employs a suitable learning algorithm and selects ads for the users tailored to the feedback given by them. These selected ads are then given to the advertisers as output. While using global differential privacy, privacy is maintained from the advertisers by ensuring that the output of the learning algorithms does not reveal information about the input (i.e., user information). Typically, advertising systems are established by leading social media networks, web browsers and other popular websites. **??** show that it is possible to accurately predict a range of highly sensitive personal attributes including age, sexual orientation, relationship status, political and religious affiliation using the feedback available to the advertising systems. Such possible breach of privacy necessitates us to protect personal user information not only from the advertisers but also from the advertising systems. Local differential privacy is able to achieve this objective unlike global differential privacy.

In this article, we propose to use *low privacy* regime using local differential privacy. In low privacy regime, the noise added to the data is small and the aim of the privacy mechanism is to send as much information about data as allowed, but no more (**?**). This is in alignment with our dual goal of using privacy in recommendation systems or Internet advertising, and other similar applications: provide useful recommendations/ads to the users while respecting their privacy as much as possible.

We measure the utility of our proposed algorithm using *regret* which is a measure of the total mistake cost (precise definitions will follow in the next Section). When rewards are bounded (as assumed in most works in the literature), the regret of any algorithm is trivially bounded linearly in the number of time steps T. An algorithm is said to be *learning* if its regret is bounded sub-linearly in T.

Main Contributions

1. We propose non-stationary stochastic corrupt bandits, a novel formulation which aims to preserve local differential privacy while still providing high utility for sequential decision making in a non-stationary environment.
2. We construct an algorithm called SW-KLUCB-CF for the considered problem.

[1] We consider a simplistic scenario for illustrative purposes.

3. We prove an upper bound on the utility of SW-KLUCB-CF in terms of its regret. This upper bound is near-optimal in terms of the number of time steps and matches the best known bound for analogous problems in terms of the number of time steps and the number of changes.
4. We provide an optimal mechanism to achieve a desired level of local differential privacy while achieving high utility.

This work is an extension of **?** to non-stationary environments and reuses some of the concepts used there. However, it should be noted that the algorithms proposed in **?** will not be able to solve the problem considered in this article. In fact, it is easy to construct non-stationary environments for which the algorithms proposed in **?** (and all other differentially private algorithms designed for stationary environment) will suffer regret linear in the number of time steps T. On the other hand, the algorithm proposed in this article can guarantee regret sub-linear in T in such scenarios. Furthermore, due to the changing environment and the use of a sliding window, the regret analysis in our article presents challenges not encountered in stationary settings.

Our extension to non-stationary environments is practically relevant as the assumption of stationarity is sometimes unrealistic in real-world applications. Such an extension providing local differential privacy in non-stationary environments for the problem of data collection is given by **?**. Our problem is different than **?** as we study learning to make optimal sequential decisions in a non-stationary environment while providing local differential privacy. Note that a naive strategy of restarting an algorithm (designed for a stationary environment) after each change is not possible in the problem considered here as the time instants at which the changes occur are unknown.

Related Work In the context of sequential decision-making, global differential privacy has been studied in various settings including stochastic bandits (**??**), adversarial bandits (**??**) and collaborative bandits (**?**). In the context of sequential decision-making, local differential privacy has been considered in stochastic bandit setting (**??**), contextual bandits (**?**), collaborative bandits (**?**) and Markov decision processes (**??**). For a comprehensive overview of differential privacy and its application to other problems, see **?**.

The notion of using a sliding window mechanism (as we do in our proposed algorithm) to deal with a non-stationary environment has been employed in classical bandits (**?**) as well as Markov decision processes (**?**).

Non-Stationary Stochastic Corrupt Bandits

A non-stationary stochastic corrupt bandits problem is formally characterized by a set of arms $A = \{1, \ldots, K\}$ on which are indexed a list of unknown sub-Gaussian reward distributions $\{\boldsymbol{\nu}_a(1)\}_{a\in A}, \ldots, \{\boldsymbol{\nu}_a(L_T)\}_{a\in A}$, a list of unknown sub-Gaussian feedback distributions $\{\boldsymbol{\varsigma}_a(1)\}_{a\in A}, \ldots, \{\boldsymbol{\varsigma}_a(L_T)\}_{a\in A}$, and a list of known *mean-corruption functions* $\{g_a\}_{a\in A}$. Here, the total number of time steps (i.e., the *horizon*) is indicated as T. The environment undergoes L_T abrupt changes at unknown time steps called as *breakpoints* and it remains constant in the intervals between two successive breakpoints.

For notational convenience, we assume that the first breakpoint occurs at $t = 1$. From i^{th} breakpoint till the subsequent breakpoint (or the horizon, in case of the last breakpoint), if the learner pulls an arm $a \in A$ at time t, they receive a (hidden) reward R_t drawn from the distribution $\boldsymbol{\nu}_a(i)$ with mean $\mu_a(i)$ and observe a feedback F_t drawn from the distribution $\boldsymbol{\varsigma}_a(i)$ with mean $\lambda_a(i)$. We assume that, for each arm, there exists a loose link between the reward and the feedback through a known *corruption function* g_a which maps the mean of the reward distribution to the mean of the feedback distribution : $g_a(\mu_a(i)) = \lambda_a(i), \forall a \in A$ and $1 \leq i \leq L_T$. Our proposed algorithm and the proven regret bound also work if the corruption function for an arm changes across time as long as the current corruption function is known.

Note that these g_a functions may be completely different from one arm to another. For Bernoulli distributions, the reward distributions and the feedback distributions are in $[0, 1]$ for all $a \in A$ and we assume all the corruption functions $\{g_a\}_{a\in A}$ to be continuous in this interval. We also assume the corruption functions $\{g_a\}_{a\in A}$ to be strictly monotonic and denote the corresponding inverse functions by g_a^{-1}. The assumption of monotonicity is required for efficient learning as proved in **?**.

Another way to define the link between the reward and the feedback is to provide a *corruption scheme* operator $\tilde{g}_a$ which maps the rewards into feedback distributions.

Randomized Response Randomized response (a privacy protection technique introduced by (**?**)) can also be simulated by a Bernoulli corrupt bandits problem and the corresponding corruption scheme $\tilde{g}_a$ is encoded as:

$$\mathbb{M}_a \coloneqq \begin{matrix} 0 \\ 1 \end{matrix} \overset{\begin{matrix} 0 & \quad 1 \end{matrix}}{\begin{bmatrix} p_{00}(a) & 1 - p_{11}(a) \\ 1 - p_{00}(a) & p_{11}(a) \end{bmatrix}} \tag{1}$$

Each item in $\mathbb{M}_a$ denotes the probability of observing a particular feedback for a particular reward i.e., $\mathbb{M}_a(y, x) \coloneqq \mathbb{P}\big(\text{Feedback from arm } a = y \mid \text{Reward from arm } a = x\big)$. The corresponding corruption function is $g_a(x) = 1 - p_{00}(a) + [p_{00}(a) + p_{11}(a) - 1] \cdot x$.

To measure the utility of an algorithm for this problem, we define the notion of regret in the following. Let us denote the mean reward of arm a at time step t as $\mu_{a,t}$. The objective of an algorithm, which chooses the arm $\hat{a}_t$ at time t based only on the previously observed feedback, $F_1, \ldots, F_{t-1}$, is to maximize the expected sum of rewards i.e., to achieve high utility. This is equivalent to minimizing the regret, $\text{Regret}(T) \coloneqq \sum_{t=1}^{T} \mu_{*,t} - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\hat{a}_t,t}\right]$, where $\mu_{*,t} \coloneqq \max_{a\in A} \mu_{a,t}$. Regret measures the performance of the algorithm against an omniscient policy that at each time step chooses the arm with the maximal mean reward. Thus, low regret translates to achieving high utility.

The Proposed Algorithm

To solve the problem at hand, we propose SW-KLUCB-CF, an adaptation of the kl-UCB algorithm of **?**. The algo-

rithm takes as input: the window size w, a non-decreasing function f, the horizon T and the corruptions functions $g_1, \ldots, g_K$. We assume that the horizon T is known; an unknown T can be handled using the doubling trick (**?**). We use $d(x, y)$ to denote the KullbackLeibler divergence between two Bernoulli distributions with mean x and y. We also use a shorthand of $x \wedge y$ to denote $\min(x, y)$.

At each time time step t, the algorithm computes an $\text{Index}_a(t)$, which is an upper-confidence bound on $\mu_{a,t}$ built from a confidence interval on $\lambda_{a,t}$ based on the KL-divergence. The quantity $N_a(t, w)$ denotes the number of times arm a was chosen in the last w time steps until time t. Correspondingly, $\hat{\lambda}_a(t, w)$ denotes the empirical mean of the feedback observed from arm a in the last w time steps until time t: $\hat{\lambda}_a(t, w) \coloneqq \frac{1}{N_a(t,w)} \sum_{s=\min\{1,t-w+1\}}^{t} F_s \cdot \mathbb{1}_{(\hat{a}_s = a)}$.

Theorem 1 gives an upper bound on the regret of SW-KLUCB-CF. A more explicit bound is proved in the Appendix.

Theorem 1 *The regret of* SW-KLUCB-CF *using* $f(x) \coloneqq \log(x) + 3\log(\log(x))$ *and* $w = \sqrt{\frac{4eT}{L_T+4}}$ *on a Bernoulli non-stationary stochastic corrupt bandits problem with strictly monotonic and continuous corruption functions* $\{g_a\}_{a \in A}$ *at time* T *is upper-bounded by* [2]

$$\tilde{O}\left(\sum_{a \in A} \sqrt{L_T T} + \sum_{i=1}^{L_T} \sum_{a \neq a_*(i)} \frac{\log\left(\sqrt{\frac{T}{L_T}}\right)}{d(\lambda_a(i), g_a(\mu_*(i))}\right),$$

where $a_*(i)$ *and* $\mu_*(i)$ *are the optimum arm and the corresponding optimal mean respectively after* i^{th} *change and before the subsequent change.*

The lower bound on regret in terms T for classical non-stationary stochastic bandits is $\Omega(\sqrt{T})$ (**?**). Theorem 1 matches the lower bound up to logarithmic factors, so SW-KLUCB-CF has near-optimal regret guarantees in terms of the time horizon T. The best known regret upper bounds for classical non-stationary stochastic bandits (e.g., **?**) also feature logarithmic terms besides the lower bound, hence our regret bound is in line with the best known results for analogous problems. Moreover, the bound in Theorem 1 also matches the best known regret bound in terms of L_T for classical non-stationary stochastic bandits which is $O\sqrt{L_T}$.

We can use SW-KLUCB-CF on non-stationary stochastic corrupts bandits where the corruption is done via randomized response. The following corollary bounds the resulting regret.

Corollary 1 *The regret of* SW-KLUCB-CF *on a Bernoulli non-stationary stochastic corrupt bandits problem with randomized response using corruption matrices* $\{\mathbb{M}\}_{a \in A}$ *at time* T *is upper-bounded by*

$$\tilde{O}\left(\sum_{a \in A} \sqrt{L_T T} + \sum_{i=1}^{L_T} \sum_{a \neq a_*(i)} \frac{\log\left(\sqrt{\frac{T}{L_T}}\right)}{(p_{00}(a) + p_{11}(a) - 1)^2}\right).$$

[2] $\tilde{O}$ ignores logarithmic factors and constants.

Algorithm 1: Sliding Window KLUCB for Non-Stationary Stochastic Corrupt Bandits (SW-KLUCB-CF)

Input: Window size w, a non-decreasing function $f : \mathbb{N} \to \mathbb{R}$, T, monotonic and continuous corruption functions $g_1, \ldots, g_K$ and $d(x, y) \coloneqq \text{KL}(\mathcal{B}(x), \mathcal{B}(y))$,

1. **Initialization:** Pull each arm once.
2. **for** time $t = K, \ldots, T-1$ **do**
   (a) Compute for each arm $a \in A$ the quantity

$$\text{Index}_a(t) \coloneqq \max\left\{q : \; N_a(t, w) \cdot d(\hat{\lambda}_a(t, w), g_a(q)) \leq f\left(t \wedge w\right)\right\}$$

   (b) Pull arm $\hat{a}_{t+1} \coloneqq \underset{a \in A}{\text{argmax}}\, \text{Index}_a(t)$ and observe the feedback F_{t+1}.

end for

This corollary follows from Theorem 1 and Pinsker's inequality: $d(x, y) > 2(x-y)^2$. The term $(p_{00}(a)+p_{11}(a)-1)$ can be understood as the slope of the corruption function g_a.

Corruption Mechanism to Preserve Local Privacy in Non-Stationary Environment

First, let us formally define local differential privacy.

Definition 1 *(Locally differentially private mechanism) Any randomized mechanism* $\mathcal{M}$ *is* ϵ*-locally differentially private for* $\epsilon \geq 0$*, if for all* $d_1, d_2 \in Domain(\mathcal{M})$ *and for all* $S \subset Range(\mathcal{M})$*,*

$$\mathbb{P}[\mathcal{M}(d_1) \in S] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(d_2) \in S].$$

As done in **?**, a straightforward approach to achieve local differential privacy using corrupt bandits is to employ a corruption scheme on the user feedback. This is similar to how randomized response is used in data collection by **?**.

Definition 2 *(ϵ-locally differentially private bandit feedback corruption scheme) A bandit feedback corruption scheme* $\tilde{g}$ *is* ϵ*-locally differentially private for* $\epsilon \geq 0$*, if for all reward sequences* $R_{t1}, \ldots, R_{t2}$ *and* $R'_{t1} \ldots, R'_{t2}$*, and for all* $\mathcal{S} \subset Range(\tilde{g})$*,*

$$\mathbb{P}[\tilde{g}(R_{t1}, \ldots, R_{t2}) \in \mathcal{S}] \leq e^{\epsilon} \cdot \mathbb{P}[\tilde{g}(R'_{t1}, \ldots, R'_{t2}) \in \mathcal{S}].$$

When corruption is done by randomized response, local differential privacy requires that $\max_{1 \leq a \leq K}\left(\frac{p_{00}(a)}{1-p_{11}(a)}, \frac{p_{11}(a)}{1-p_{00}(a)}\right) \leq e^{\epsilon}$. From Corollary 1, we can see that to achieve lower regret, $p_{00}(a) + p_{11}$(a) is to be maximized for all $a \in A$. Using **?**, Result 1, we can state that, in order to achieve ϵ-local differential privacy while maximizing $p_{00}(a) + p_{11}(a)$,

$$\mathbb{M}_a = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} \frac{e^{\epsilon}}{1+e^{\epsilon}} & \frac{1}{1+e^{\epsilon}} \\ \frac{1}{1+e^{\epsilon}} & \frac{e^{\epsilon}}{1+e^{\epsilon}} \end{bmatrix} \end{array}. \tag{2}$$

As it turns out, this is equivalent to the *staircase* mechanism for local privacy which is the optimal local differential privacy mechanism for low privacy regime (**?**, Theorem 14). The trade-off between utility and privacy is controlled by ϵ. Using the corruption parameters from Eq. (2) with Corollary 1, we arrive at the following upper bound.

Corollary 2 *At time T, the regret of* SW-KLUCB-CF *with ϵ-locally differentially private bandit feedback corruption scheme given by Eq.* (2) *is* $\tilde{O}\left(\sum_{a\in A}\sqrt{L_T T} + \sum_{i=1}^{L_T}\sum_{a\neq a_*(i)} \frac{\log\left(\sqrt{\frac{T}{L_T}}\right)}{\left(\frac{e^\epsilon-1}{e^\epsilon+1}\right)^2}\right)$.

The term $\left(\frac{e^\epsilon-1}{e^\epsilon+1}\right)^2$ in the above expression conveys the relationship of the regret with the level of local differential privacy symbolized by ϵ. For low values of ϵ, $\left(\frac{e^\epsilon-1}{e^\epsilon+1}\right) \approx \epsilon/2$. This is in line with other bandit algorithms providing differential privacy (e.g., **?**).

Elements of Mathematical Analysis

Here, we provide a proof outline for Theorem 1. Please refer to the Appendix for the complete proof. We start by bounding the expected number of times a suboptimal arm (i.e., an arm other than the optimal arm at the time of selection) is pulled by the algorithm till horizon T. Recall that, at any time step t, SW-KLUCB-CF pulls an arm maximizing an index defined as

$$\begin{aligned}&\mathrm{Index}_a(t)\\ &\coloneqq \max\left\{q:\ N_a(t,w)\cdot d\left(\hat{\lambda}_a(t,w), g_a(q)\right) \leq f\left(t\wedge w\right)\right\}\\ &= \max g_a^{-1}\left(\left\{q: N_a(t,w)\cdot d\left(\hat{\lambda}_a(t,w), q\right) \leq f\left(t\wedge w\right)\right\}\right).\end{aligned}$$

We further decompose the computation of index as follows,

$$\mathrm{Index}_a(t) \coloneqq \begin{cases} g_a^{-1}(\ell_a(t)) & \text{if } g_a \text{ is decreasing,}\\ g_a^{-1}(u_a(t)) & \text{if } g_a \text{ is increasing}\end{cases}$$

where,

$$\ell_a(t) \coloneqq \min\left\{q: N_a(t,w)\cdot d\left(\hat{\lambda}_a(t,w), q\right) \leq f\left(t\wedge w\right)\right\},$$
$$u_a(t) \coloneqq \max\left\{q: N_a(t,w)\cdot d\left(\hat{\lambda}_a(t,w), q\right) \leq f\left(t\wedge w\right)\right\}.$$

The interval $[\ell_a(t), u_a(t)]$ is a KL-based confidence interval on the mean feedback $\lambda_{a,t}$ of arm a at time t. This is in contrast to kl-UCB (**?**) where a confidence interval is placed on the mean reward. Furthermore, This differs from kl-UCB-CF (**?**) where the mean feedback of an arm remains the same for all the time steps and f does not feature w.

In our analysis, we use the fact that when an arm a is picked at time $t+1$ by SW-KLUCB-CF, one of the following is true: Either the mean feedback of the optimal arm $a_{*,t}$ with mean reward $\mu_{*,t}$ is outside its confidence interval (i.e., $g_{a_{*,t}}(\mu_{*,t}) < \ell_{a_{*,t}}(t)$ or $g_{a_{*,t}}(\mu_{*,t}) > u_{a_{*,t}}(t)$) which is unlikely. Or, the mean feedback of the optimal arm is where it should be, and then the fact that arm a is selected indicates that the confidence interval on λ_a cannot be too small as either $(u_a(t) \geq g_a(\mu_{*,t}))$ or $(\ell_a(t) \leq g_a(\mu_{*,t}))$. The previous statement follows from considering various cases depending on whether the corruption functions g_a and $g_{a_{*,t}}$ are increasing or decreasing. We then need to control the two terms in the decomposition of the expected number of draws of arm a. The term regarding the "unlikely" event, is bounded using the same technique as in the kl-UCB analysis, however with some added challenges due to the use of a sliding window. In particular, the analysis of a typical upper confidence bound algorithm for bandits relies on the fact that the confidence interval for any arm is always non-increasing, however this is not true while using a sliding window. To control the second term, depending on the monotonicity of the corruption functions g_a and $g_{a_{*,t}}$, we need to meticulously adapt the arguments in **?** to control the number of draws of a suboptimal arm, as can be seen in the Appendix.

Concluding Remarks

In this work, we proposed the setting of non-stationary stochastic corrupt bandits for preserving privacy while still maintaining high utility in sequential decision making in a changing environment. We devised an algorithm called SW-KLUCB-CF and proved its regret upper bound which is near-optimal in the number of time steps and matches the best known bound for analogous problems in terms of the number of time steps and the number of changes. Moreover, we provided an optimal corruption scheme to be used with our algorithm in order to attain the dual goal of achieving high utility while maintaining the desired level of privacy.

Interesting directions for future work include:

1. Complete an empirical evaluation of the proposed algorithm on simulated as well as real-life data.
2. Characterize the changes in the environment by a variation budget (as done in **?** for classical bandits) instead of the number of changes.
3. Incorporate contextual information in the learning process.
4. Propose a Bayesian algorithm for non-stationary stochastic corrupt bandits.
5. Propose a (near-)optimal differentially private algorithm which does not need to know the number of changes.

Quo tenetur repudiandae incidunt minus, consequuntur esse pariatur fuga facilis neque, quaerat accusantium quidem eligendi amet alias mollitia, dolor esse at deleniti quas voluptas officiis reprehenderit ut hic.Cupiditate cumque recusandae sequi incidunt nemo similique sit reiciendis, consequuntur ut labore possimus vero eveniet cumque at aliquid laboriosam eligendi atque.Illo vel omnis debitis aliquam optio quisquam a dignissimos ipsa repellendus nemo, ad sint nobis ea a rerum accusamus consequatur placeat tempore, facere ipsam iusto et eum fuga, natus doloribus quaerat doloremque temporibus cupiditate hic?Illo veniam cumque porro, fuga quas quo pariatur repellat, laboriosam impedit inventore ab architecto officiis neque iste cumque aperiam iusto culpa, eaque quia ab repellendus sapiente.Quibusdam ipsam tempore dolorum, accusamus error animi iusto quam nisi maxime, eaque beatae doloribus, ratione hic animi consequatur sit recusandae eius quo?Corrupti facere numquam

omnis illum nam eius nobis, in esse dolorum, praesentium veniam ab sint atque accusantium tempore?Cum esse vero alias obcaecati exercitationem corporis sed