An Efficient Algorithm For Generalized Linear Bandit: Online Stochastic Gradient Descent and Thompson Sampling

Qin Ding
Department of Statistics
University of California, Davis
qding@ucdavis.edu

Cho-Jui Hsieh
Department of Computer Science
University of California, Los Angeles
chohsieh@cs.ucla.edu

James Sharpnack
Department of Statistics
University of California, Davis
jsharpna@ucdavis.edu

Abstract

We consider the contextual bandit problem, where a player sequentially makes decisions based on past observations to maximize the cumulative reward. Although many algorithms have been proposed for contextual bandit, most of them rely on finding the maximum likelihood estimator at each iteration, which requires $O(t)$ time at the t-th iteration and are memory inefficient. A natural way to resolve this problem is to apply online stochastic gradient descent (SGD) so that the per-step time and memory complexity can be reduced to constant with respect to t, but a contextual bandit policy based on online SGD updates that balances exploration and exploitation has remained elusive. In this work, we show that online SGD can be applied to the generalized linear bandit problem. The proposed SGD-TS algorithm, which uses a single-step SGD update to exploit past information and uses Thompson Sampling for exploration, achieves $\tilde{O}(\sqrt{T})$ regret with the total time complexity that scales linearly in T and d, where T is the total number of rounds and d is the number of features. Experimental results show that SGD-TS consistently outperforms existing algorithms on both synthetic and real datasets.

1 INTRODUCTION

A contextual bandit is a sequential learning problem, where each round the player has to decide which action to take by pulling an arm from K arms. Before making the decisions at each round, the player is given the information of K arms, represented by d-dimensional feature vectors. Only the rewards of pulled arms are revealed to the player and the player may use past observations to estimate the relationship between feature vectors and rewards. However, the reward estimate is biased towards the pulled arms as the player cannot observe the rewards of unselected arms. The goal of the player is to maximize the cumulative reward or minimize cumulative regret across T rounds. Due to this partial feedback setting in bandit problems, the player is facing a dilemma of whether to exploit by pulling the best arm based on the current estimates, or to explore uncertain arms to improve the reward estimates. This is the so-called exploration-exploitation trade-off. Contextual bandit problem has substantial applications in recommender system (Li et al., 2010), clinical trials (Woodroofe, 1979), online advertising (Schwartz et al., 2017), etc. It is also the fundamental problem of reinforcement learning (Sutton et al., 1998).

The most classic problem in contextual bandit is the stochastic linear bandit (Abbasi-Yadkori et al., 2011; Chu et al., 2011), where the expected rewards follow a linear model of the feature vectors and an unknown model parameter $\theta^* \in \mathbb{R}^d$. Upper Confidence Bound (UCB) (Abbasi-Yadkori et al., 2011; Auer et al., 2002; Chu et al., 2011) and Thompson Sampling (TS) (Thompson, 1933; Agrawal and Goyal, 2012, 2013; Chapelle and Li, 2011) are two most popular algorithms to solve bandit problems. UCB uses the upper

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

confidence bound to estimate the reward optimistically and therefore mixes exploration into exploitation. TS assumes the model parameter follows a prior and uses a random sample from the posterior to estimate the reward model. Despite the popularity of stochastic linear bandit, linear model is restrictive in representation power and the assumption of linearity rarely holds in practice. This leads to extensive studies in more complex contextual bandit problems such as generalized linear bandit (GLB) (Filippi et al., 2010; Jun et al., 2017; Li et al., 2017), where the rewards follow a generalized linear model (GLM). Li et al. (2012) shows by extensive experiments that GLB achieves lower regret than linear bandit in practice.

For most applications of contextual bandit, efficiency is crucial as the decisions need to be made in real time. While GLB can still be solved by UCB or TS, the estimate of upper confidence bound or posterior becomes much more challenging than the linear case. It does not have closed form in general and has to be approximated, which usually requires costly operations in online learning. As pointed out by Li et al. (2017), most GLB algorithms suffer from two expensive operations. The first is that they need to invert a $d \times d$ matrix every round, which is time-consuming when d is large. The second is that they need to find the maximum likelihood estimator (MLE) by solving an optimization problem using all the previous observations at each round. This results in $\Omega(T^2)$ time and $O(T)$ memory for T rounds.

From an optimization perspective, stochastic gradient descent (SGD) (Hazan et al., 2016) is a popular algorithm for both convex and non-convex problems, even for complex models like neural networks. Online SGD (Hazan et al., 2016) is an efficient optimization algorithm that incrementally updates the estimator via new observations at each round. Although it is natural to apply online SGD to contextual bandit problems so that the time complexity at the t-th round can be reduced to constant with respect to t, it has not been successfully used due to the following reasons: 1) the hardness of constructing unbiased stochastic gradient with controllable variance due to the partial feedback setting in bandit problems, 2) the difficulty to achieve a balance between sufficient exploration and fast convergence to the optimal decision using solely online SGD, 3) lack of theoretical guarantee. Previous attempts of online SGD in contextual bandit problems are limited to empirical studies. Bietti et al. (2018) uses importance weight and doubly-robust techniques to construct unbiased stochastic gradient with reduced variance. In Riquelme et al. (2018), it is shown that the inherit randomness of SGD does not always offer enough exploration for bandit problems. To the best of our knowledge, there is no existing work that can successfully apply online SGD to update the model parameter of a contextual bandit, while maintaining low theoretical regret.

In this work, we study how online SGD can be appropriately applied to GLB problems. To overcome the dilemma of exploration and exploitation, we propose an algorithm that carefully combines online SGD and TS techniques for GLB. The exploration factor in TS is re-calibrated to make up for the gap between SGD estimator and MLE. Interestingly, we found that by doing so, we can skip the step of inverting matrices. This leads to $O(Td)$ time complexity of our proposed algorithm when T is much bigger than d, which is the most efficient GLB algorithm so far. We provide theoretical guarantee of our algorithm and show that under the "diversity" assumption (formally defined in Assumption 3 of Section 3), it can obtain $\tilde{O}(\sqrt{T})$[1] regret upper bound for finite-arm GLB problems. Recently, similar "diversity" assumptions have been made to analyze the regret bounds of linear UCB (LinUCB) (Wu et al., 2020), greedy algorithms (Bastani et al., 2020; Kannan et al., 2018) or perturbed adversarial bandit setting (Kannan et al., 2018), though none of them improve the efficiency of contextual bandit algorithms, which is one of the most important contributions of our work. We will discuss in Remark 1 the comparisons of previous "diversity" assumptions and ours.

Notations: We use θ^* to denote the true model parameter. For a vector $x \in \mathbb{R}^d$, we use $\|x\|$ to denote its l_2 norm and $\|x\|_A = \sqrt{x^T A x}$ to denote its weighted l_2 norm associate with a positive-definite matrix $A \in \mathbb{R}^{d\times d}$. We use $\lambda_{\min}(A)$ to denote the minimum eigenvalue of a matrix A. Denote $[n] := \{1, 2, \dots, n\}$ and f' as the first derivative of a function f. Finally, we use $\lfloor b \rfloor$ to denote the maximum integer such that $\lfloor b \rfloor \le b$ and use $\lceil b \rceil$ to denote the minimum integer such that $\lceil b \rceil \ge b$.

2 RELATED WORK

In this section, we briefly discuss some previous algorithms in GLB. Filippi et al. (2010) first proposes a UCB type algorithm, called GLM-UCB. It achieves $\tilde{O}(\sqrt{T})$ regret upper bound. According to Dani et al. (2008), this regret bound is optimal up to logarithmic factors for contextual bandit problems. Li et al. (2017) proposes a similar algorithm called UCB-GLM. It improves the regret bound of GLM-UCB by a $\sqrt{\log T}$ factor. The main idea is to calculate the MLE of θ^* at each round, and then find the upper confidence bound of reward estimates. The time complexity of these two

[1] $\tilde{O}$ ignores poly-logarithmic factors.

algorithms depends quadratically on both d and T as they need to calculate the MLE and matrix inverse every round. SupCB-GLM (Li et al., 2017) has similar regret bounds for finite-arm GLB problem. Its theoretical time complexity is similar to UCB-GLM, although it is impractical generally.

Another rich line of algorithms for GLB follows TS scheme, where the key is to estimate the posterior of θ^* after observing extra data at each round. Laplace-TS (Chapelle and Li, 2011) estimates the posterior of regularized logistic regression by Laplace approximations, whose per-round time complexity is $O(d)$. However, Laplace-TS works only for logistic bandit and does not apply to general GLB problems. Moreover, it performs poorly when the feature vectors are non-Gaussian and when $d > K$. Dumitrascu et al. (2018) proposes Pólya-Gamma augmented Thompson Sampling (PG-TS) with a Gibbs sampler to estimate the posterior for logistic bandit. However, Gibbs sampler inference is very expensive in online algorithms. The time complexity of PG-TS is $O(M(d^2T^2 + d^3T))$, where M is the burn-in step. In general, previous TS based algorithms for logistic bandit have regret bound $\tilde{O}(\sqrt{T})$ (Dong et al., 2019; Abeille et al., 2017; Russo and Van Roy, 2014).

More recently, Kveton et al. (2020) proposed two algorithms for GLB, both enjoy $\tilde{O}(\sqrt{T})$ total regret. GLM-TSL (Kveton et al., 2020) follows the TS technique. It draws a sample from the approximated posterior distribution and pulls the arm with the best estimates of this posterior. As it needs to calculate the MLE and the covariance matrix of the posterior needs to be reweighted using previous pulls every round, its time complexity depends quadratically on both d and T. GLM-FPL (Kveton et al., 2020) fits a generalized linear model to the past rewards randomly perturbed by the Gaussian noises and pulls the arm that has the best reward based on this model. Its time complexity is also quadratic on T.

In addition to UCB and TS algorithm, ϵ-greedy algorithm (Auer et al., 2002; Sutton et al., 1998) is also very popular in practice due to its simplicity, although it does not have theoretical guarantee in general bandit framework. At each round, ϵ-greedy has probability ϵ to randomly pull an arm, and has probability $1 - \epsilon$ to pull the best arm from the current estimates. The time complexity of ϵ-greedy algorithm depends quadratically on T as it need to calculate the MLE every round to find the current best estimates.

To make GLB algorithms scalable, Jun et al. (2017) proposes Generalized Linear Online-to-confidence-set Conversion (GLOC) algorithm. GLOC utilizes the exp-concavity of the loss function of GLM and applies online Newton steps to construct a confidence set for θ^*. GLOC and its TS version, GLOC-TS both achieve $\tilde{O}(\sqrt{T})$ regret upper bound. The total time complexity of GLOC is $O(Td^2)$ due to the successful use of an online second order update. However, GLOC remains expensive when d is large. We show a detailed analysis of time complexity of GLB algorithms in Table 1 of Section 6.

3 PROBLEM SETTING

We consider the K-armed stochastic generalized linear bandit (GLB) setting. Denote T as the total number of rounds. At each round $t \in [T]$, the player observes a set of contexts including K feature vectors $\mathcal{A}_t := \{x_{t,a} | a \in [K]\} \subset \mathbb{R}^d$. $\mathcal{A}_t$ is drawn IID from an unknown distribution with $\|x_{t,a}\| \leq 1$ for all $t \in [T]$ and $a \in [K]$, where $x_{t,a}$ represents the information of arm a at round t. We make the same regularity assumption as in Li et al. (2017), i.e., there exists a constant $\sigma_0 > 0$ such that $\lambda_{\min}\left(\mathbb{E}\left[\frac{1}{K}\sum_{a=1}^{K} x_{t,a}x_{t,a}^T\right]\right) \geq \sigma_0^2$. Denote $y_{t,a}$ as the associated random reward of arm a at round t. After $\mathcal{A}_t$ is revealed to the player, the player pulls an arm $a_t \in [K]$ and only observes the reward associated with the pulled arm, y_{t,a_t}. In the following, we denote $Y_t = y_{t,a_t}$ and $X_t = x_{t,a_t}$.

In GLB, the expected rewards follow a generalized linear model (GLM) of the feature vectors and an unknown vector $\theta^* \in \mathbb{R}^d$, i.e., there is a fixed, strictly increasing link function $\mu : \mathbb{R} \to \mathbb{R}$ such that $\mathbb{E}[y_{t,a}|x_{t,a}] = \mu(x_{t,a}^T\theta^*)$ for all t and a. For example, linear bandit and logistic bandit are special cases of GLB with $\mu(x) = x$ and $\mu(x) = 1/(1 + e^{-x})$ respectively. Without loss of generality, we assume $\mu(x) \in [0, 1]$ and $y_{t,a} \in [0, 1]$.[2] We also assume that Y_t follows a sub-Gaussian distribution with parameter $R > 0$. Formally, the GLM can be written as $Y_t = \mu(X_t^T\theta^*) + \epsilon_t$, where ϵ_t are independent zero-mean sub-Gaussian noises with parameter R. We use $\mathcal{F}_t = \sigma(a_1, \ldots, a_t, \mathcal{A}_1, \ldots, \mathcal{A}_t, Y_1, \ldots, Y_t)$ to denote the σ-algebra generated by all the information up to round t. Then we have $\mathbb{E}\left[e^{\lambda\epsilon_t}|\mathcal{F}_{t-1}\right] \leq e^{\frac{\lambda^2R^2}{2}}$ for all t and $\lambda \in \mathbb{R}$. Denote $a_t^* = \text{argmax}_{a\in[K]}\, \mu(x_{t,a}^T\theta^*)$ and $x_{t,*} = x_{t,a_t^*}$, the cumulative regret of T rounds is defined as

$$R(T) = \sum_{t=1}^{T} \left[\mu(x_{t,*}^T\theta^*) - \mu(X_t^T\theta^*)\right]. \tag{1}$$

The player's goal is to find an optimal policy π, such that if the player follows policy π to pull arm a_t at round t, the total regret $R(T)$ or the expected regret

[2] Rewards in $[0, 1]$ is a non-critical assumption, which can be easily removed. In fact, we only need the rewards to have bounded variance for all the analysis to work.

$\mathbb{E}[R(T)]$ is minimized. Note that $R(T)$ is random due to the randomness in a_t. We make the following mild assumptions similar to Li et al. (2017).

Assumption 1. *μ is differentiable and there exists a constant $L_\mu > 0$ such that $|\mu'| \le L_\mu$.*

For logistic link function, Assumption 1 holds when $L_\mu = \frac{1}{4}$. For linear function, we have $L_\mu = 1$.

Assumption 2. *We assume $c_3 > 0$, where $c_\eta := \inf_{\{\|x\|\le 1, \|\theta-\theta^*\|\le \eta\}} \mu'(x^T\theta)$.*

This assumption is not stronger than the assumption made in Li et al. (2017) for linear bandit and logistic bandit, as Li et al. (2017) assumes $c_1 > 0$ and $\frac{c_3}{c_1} \sim O(1)$ in both cases.

To make sure we can successfully apply online SGD update in bandit problems, we also need the following regularity assumption, which assumes that the optimal arm based on any model parameter θ has non-singular second moment matrix. This assumption is similar to the regularity assumption made in Li et al. (2017), which assumes that the averaged second moment matrices of feature vectors, i.e., $\mathbb{E}[\frac{1}{K}\sum_{a=1}^K x_{t,a}x_{t,a}^T]$ is non-singular. Assumption 3 below merely says that the same holds for the optimal arm based on any θ.

Assumption 3. *For a fixed $\theta \in \mathbb{R}^d$, let $\tilde{X}_{\theta,t} = \text{argmax}_{a\in[K]}\, \theta^T x_{t,a}$ Denote $\Sigma_\theta = \mathbb{E}[\tilde{X}_{\theta,t}\tilde{X}_{\theta,t}^T]$ and $\lambda_f = \inf_\theta \lambda_{\min}(\Sigma_\theta)$. We assume λ_f is a positive constant.*

Intuitively, Assumption 3 means that based on any model parameter θ, the projection of the optimal arm's feature vector onto any direction has positive probability to be non-zero. In practice, the optimal arms at different rounds are diverse, so it is reasonable to assume that the projections of these random vectors onto any direction are not always a constant zero.

Remark 1. *Wu et al. (2020) makes another version of diversity assumption and proposes the LinUCB-d algorithm to utilize the diversity property of contexts. It requires that all arms could be optimal under certain contexts and that the corresponding feature vectors span $\mathbb{R}^d$. Our Assumption 3 is different from the one in Wu et al. (2020) since we do not require that all the arms could be optimal. Moreover, LinUCB-d only works for linear bandit and cannot be generalized to GLB problems easily. Even in the linear case, the time complexity of LinUCB-d depends quadratically on d. Bastani et al. (2020) analyzes the greedy algorithm under a diversity assumption, which assumes the covariance matrix of all the feature vectors lying in any half space is positive definite. Our assumption is different from this since we only make the diversity assumption on the optimal arm under different θ, instead of all the feature vectors. We will include the experimental comparisons with ϵ-greedy algorithms for GLB problems in Section 6 and show that our algorithm significantly outperforms it.*

4 PROPOSED ALGORITHM

In this section, we formally describe our proposed algorithm. The main idea is to use online stochastic gradient descent (SGD) procedure to estimate the MLE and use Thompson Sampling (TS) to explore.

For GLM, the MLE from n data points $\{X_i, Y_i\}_{i=1}^n$ is $\hat{\theta}_n = \text{argmax}_\theta \sum_{i=1}^n \left[Y_i X_i^T\theta - m(X_i^T\theta)\right]$, where $m'(x) = \mu(x)$. Therefore, it is natural to define the loss function at round t to be $l_t(\theta) = -Y_t X_t^T\theta + m(X_t^T\theta)$. Effective algorithms in GLB (Abeille et al., 2017; Filippi et al., 2010; Li et al., 2017; Russo and Van Roy, 2014) have been shown to converge to the optimal action at a rate of $\tilde{O}(\frac{1}{\sqrt{T}})$. Similarly, we need to ensure that online SGD steps will achieve the same fast convergence rate. This rate is only attainable when the loss function is strongly convex. However, the loss function at a single round is convex but not necessarily strongly convex. To tackle this problem, we aggregate the loss function every τ steps, where τ is a parameter to be specified. We define the j-th aggregated loss function as

$$l_{j,\tau}(\theta) = \sum_{s=(j-1)\tau+1}^{j\tau} -Y_s X_s^T\theta + m(X_s^T\theta). \qquad (2)$$

Let α be a positive constant, we will show in Section 5 that when τ is appropriately chosen based on α, the aggregated loss function of τ rounds is α-strongly convex and therefore fast convergence can be obtained. The gradient and Hessian of $l_{j,\tau}$ are derived as

$$\nabla l_{j,\tau}(\theta) = \sum_{s=(j-1)\tau+1}^{j\tau} -Y_s X_s + \mu(X_s^T\theta)X_s, \qquad (3)$$

$$\nabla^2 l_{j,\tau}(\theta) = \sum_{s=(j-1)\tau+1}^{j\tau} \mu'(X_s^T\theta)X_s X_s^T. \qquad (4)$$

In the first τ rounds of the algorithm, we randomly pull arms. Denote $\hat{\theta}_t$ as the MLE at round t using previous t observations. We calculate the MLE only once at round τ and get $\hat{\theta}_\tau$. We keep a convex set $\mathcal{C} = \{\theta : \|\theta - \hat{\theta}_\tau\| \le 2\}$. We will show in Section 5 that when τ is properly chosen, we have $\|\hat{\theta}_t - \theta^*\| \le 1$ for all $t \ge \tau$. Therefore, for every $t \ge \tau$, we have $\hat{\theta}_t \in \mathcal{C}$. Denote $\tilde{\theta}_j$ as the j-th updated SGD estimator and let $\tilde{\theta}_0 = \hat{\theta}_\tau$. Starting from round $t = \tau + 1$, we update $\tilde{\theta}_j$ every τ rounds. Since the minimum of the loss function lies in $\mathcal{C}$, we project $\tilde{\theta}_j$ to the convex set $\mathcal{C}$ (line 9 of Algorithm 1). Define $\bar{\theta}_j = \frac{1}{j}\sum_{q=1}^j \tilde{\theta}_q$, then $\bar{\theta}_j$

is treated as the posterior mean of θ^* and we use TS to ensure sufficient exploration. Specifically, we draw θ_j^{TS} from a multivariate Gaussian distribution with mean $\bar{\theta}_j$ and covariance matrix

$$A_j = \left(\frac{2c_3 g_1(j)^2}{\alpha j} + \frac{2g_2(j)^2}{j} \right) I_d, \qquad (5)$$

where $g_1(j)$ and $g_2(j)$ are defined as

$$g_1(j) = \frac{R}{c_1} \sqrt{\frac{d}{2} \log(1 + \frac{2j\tau}{d}) + 2 \log T} \qquad (6)$$

$$g_2(j) = \frac{\tau}{\alpha} \sqrt{1 + \log j}. \qquad (7)$$

Previous works (Filippi et al., 2010; Li et al., 2017; Jun et al., 2017) in GLB use V_{t+1}^{-1} as the covariance matrix, where $V_{t+1} = \sum_{s=1}^{t} X_s X_s^T$. In contrast, we use $\frac{2c_3 g_1(j)^2}{\alpha j} I_d$ to approximate $V_{j\tau+1}^{-1}$. Meanwhile, the covariance matrix in Equation 5 has an extra second term, which comes from the gap between the averaged SGD estimator $\bar{\theta}_j$ and the MLE $\hat{\theta}_{j\tau}$. Note that similar to the SGD estimator $\tilde{\theta}_j$, TS estimator θ_j^{TS} is updated every τ rounds. At round $t > \tau$, we will pull arm $a_t = \text{argmax}_{a \in [K]} \mu(x_{t,a}^T \theta_j^{\text{TS}})$, where $j = \lfloor \frac{t-1}{\tau} \rfloor$. See Figure 1 for a brief illustration of the notations. Since our proposed algorithm employs both techniques from online SGD and TS methods, we call our algorithm SGD-TS. See Algorithm 1 for details.

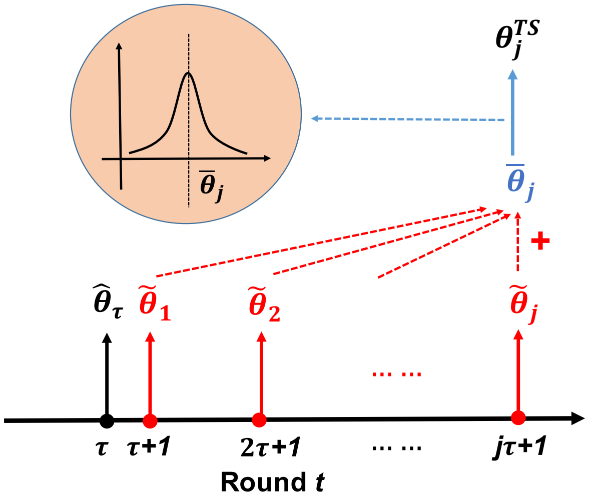

Figure 1: Illustration of notations.

Since some GLB algorithms like UCB-GLM (Li et al., 2017) and GLM-UCB (Filippi et al., 2010) need to compute MLE every round, to be able to compare the time complexity, we assume the MLE using t datapoints with d features can be solved in $O(td)$ time. SGD-TS is an extremely efficient algorithm for GLB. We only calculate the MLE once at the τ-th round, which costs $O(\tau d)$ time. Then we update the SGD estimator every τ rounds and the gradient can be incrementally computed with per-round time $O(d)$. Note that we do not need to calculate matrix inverse every round either since we approximate V_{t+1}^{-1} by a diagonal matrix. In conclusion, the time complexity of SGD-TS in T rounds is $O(Td + d\tau)$, and it will be shown in Section 5 that $\tau \sim O(\max\{d, \log T\}/\lambda_f^2)$. In practice, T is usually much greater than d, and in such cases, SGD-TS costs $O(Td)$ time. Our algorithm improves the efficiency significantly if either d or T is large. See Table 1 in Section 6 for comparisons with other algorithms.

Algorithm 1 Online stochastic gradient descent with Thompson Sampling (SGD-TS)

```
Input: $T, K, \tau, \alpha$.
Randomly choose $a_t \in [K]$ and record $X_t$, $Y_t$ for $t \in [\tau]$.
Calculate the maximum-likelihood estimator $\hat{\theta}_\tau$ by solving $\sum_{t=1}^{\tau} (Y_t - \mu(X_t^T \theta)) X_t = 0$.
Maintain convex set $\mathcal{C} = \{\theta : \|\theta - \hat{\theta}_\tau\| \le 2\}$.
$\tilde{\theta}_0 \leftarrow \hat{\theta}_\tau$.
for $t = \tau + 1$ to $T$ do
    if $t \% \tau = 1$ then
        $j \leftarrow \lfloor (t-1)/\tau \rfloor$ and $\eta_j = \frac{1}{\alpha j}$.
        Calculate $\nabla l_{j,\tau}$ defined in Equation 3
        Update $\tilde{\theta}_j \leftarrow \prod_{\mathcal{C}} \left( \tilde{\theta}_{j-1} - \eta_j \nabla l_{j,\tau}(\tilde{\theta}_{j-1}) \right)$.
        Compute $\bar{\theta}_j = \frac{1}{j} \sum_{q=1}^{j} \tilde{\theta}_q$.
        Compute $A_j$ defined in Equation 5.
        Draw $\theta_j^{\text{TS}} \sim \mathcal{N}(\bar{\theta}_j, A_j)$.
    end if
    Pull arm $a_t \leftarrow \text{argmax}_{a \in [K]} \mu(x_{t,a}^T \theta_j^{\text{TS}})$ and observe reward $Y_t$.
end for
```

5 MATHEMATICAL ANALYSIS

In this section, we formally analyze Algorithm 1. Proofs are deferred to supplementary materials.

5.1 Convergence of SGD update

Lemma 1. *Denote* $V_{t+1} = \sum_{s=1}^{t} X_s X_s^T$. *If* $\lambda_{\min}(V_{t+1}) \ge \frac{16R^2[d + \log(\frac{1}{\delta_1})]}{c_1^2}$, *where* δ_1 *is a small probability, then* $\|\hat{\theta}_t - \theta^*\| \le 1$ *holds with probability at least* $1 - \delta_1$.

From Lemma 1, we have $\hat{\theta}_t \in \mathcal{C}$ with probability at least $1 - \delta_1$ when $t \ge \tau$ as long as τ is properly chosen. This is essential because the SGD estimator is projected to $\mathcal{C}$. In Lemma 2, we show that when τ is chosen as

Equation 8, the averaged SGD estimator $\bar{\theta}_j$ converges to MLE at a rate of $\tilde{O}(\frac{1}{\sqrt{j}})$.

Lemma 2. *For a constant $\alpha > 0$, let*

$$\begin{aligned}\tau_1 &= \left(\frac{C_1\sqrt{d} + C_2\sqrt{2\log T}}{\sigma_0^2}\right)^2 + \frac{32R^2[d + 2\log T]}{c_1^2\sigma_0^2},\\ \tau_2 &= \left(\frac{C_1\sqrt{d} + C_2\sqrt{3\log T}}{\lambda_f}\right)^2 + \frac{2\alpha}{c_3\lambda_f},\\ \tau &= \lceil\max\{\tau_1, \tau_2\}\rceil, \end{aligned} \tag{8}$$

where C_1 and C_2 are two universal constants, then with probability at least $1 - \frac{3}{T^2}$, the following holds when $j \geq 1$,

$$\|\bar{\theta}_j - \hat{\theta}_{j\tau}\| \leq \frac{\tau}{\alpha}\sqrt{\frac{1 + \log j}{j}}.$$

5.2 Concentration events

By the property of MLE and Lemma 2, we have the concentration property of SGD estimator.

Lemma 3. *Suppose τ is chosen as in Equation 8, and $\alpha \geq c_3$, define $\mathbb{B}_1^d = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$, we have $E_1(j)$ holds with probability at least $1 - \frac{5}{T^2}$, where $E_1(j) = \{x \in \mathbb{B}_1^d : |x^T(\bar{\theta}_j - \theta^*)| \leq g_1(j)\|x\|_{V_{j\tau+1}^{-1}} + g_2(j)\frac{\|x\|}{\sqrt{j}}\}$ and $g_1(j)$ and $g_2(j)$ are defined in Equation 6 and Equation 7.*

The following lemma shows the concentration property of TS estimator.

Lemma 4. *Define $u = \sqrt{2\log(K\tau T^2)}$, we have $\mathbb{P}(E_2(j)|\mathcal{F}_{j\tau}) \geq 1 - \frac{1}{T^2}$, where $E_2(j)$ is defined as the set of all the vectors $x \in \left\{\cup_{t=j\tau+1}^{(j+1)\tau}\mathcal{A}_t\right\}$ such that the following inequality holds*

$$|x^T(\bar{\theta}_j - \theta_j^{TS})| \leq u\sqrt{\frac{2c_3 g_1(j)^2}{\alpha j}\|x\|^2 + 2g_2(j)^2\frac{\|x\|^2}{j}}.$$

The above two lemmas show that the TS estimator θ_j^{TS} is concentrated around the true model parameter θ^*. Lemma 5 below offers the anti-concentration property of TS estimator, which ensures that we have enough exploration for the optimal arm.

Lemma 5. *Denote $j_t = \lfloor\frac{t-1}{\tau}\rfloor$. For any filtration $\mathcal{F}_t$ such that $E_1(j_t) \cap \{\lambda_{\min}(V_{j_t\tau+1}) \geq \frac{\alpha j_t}{c_3}\}$ is true, we have $\mathbb{P}\left(x_{t,*}^T\theta_{j_t}^{TS} > x_{t,*}^T\theta^*|\mathcal{F}_{j_t\tau}\right) \geq \frac{1}{4\sqrt{\pi e}}$.*

5.3 Regret analysis

Using the concentration and anti-concentration properties of TS estimator in Lemma 3, 4 and 5, we are able to bound a single-round regret in Lemma 6. Denote $\Delta_i(t) = (x_{t,*} - x_{t,i})^T\theta^*$, $j_t = \lfloor\frac{t-1}{\tau}\rfloor$ and

$$\begin{aligned}H_i(t) = {} & g_1(j_t)\|x_{t,i}\|_{V_{j_t\tau+1}^{-1}} + g_2(j_t)\frac{\|x_{t,i}\|}{\sqrt{j_t}}\\ & + u\sqrt{\frac{2c_3 g_1(j_t)^2}{\alpha j_t}\|x_{t,i}\|^2 + 2g_2(j_t)^2\frac{\|x_{t,i}\|^2}{j_t}}. \end{aligned} \tag{9}$$

Lemma 6. *At round $t \geq \tau$, where τ is defined in Equation 8, denote $E_3(j_t) = \{\lambda_{\min}(V_{j_t\tau+1}) \geq \frac{\alpha j_t}{c_3}\}$, we have*

$$\begin{aligned}&\mathbb{E}[\Delta_{a_t}(t)\mathbb{1}(E_1(j_t) \cap E_2(j_t) \cap E_3(j_t))]\\ &\leq \left(1 + \frac{2}{\frac{1}{4\sqrt{\pi e}} - \frac{1}{T^2}}\right)\mathbb{E}\left[H_{a_t}(t)\mathbb{1}(E_3(j_t))\right]. \end{aligned} \tag{10}$$

We are now ready to put together the above information and prove the regret bound of Algorithm 1.

Theorem 1. *When Algorithm 1 runs with $\alpha = \max\{c_3, d, \log T\}/\lambda_f$, and τ defined in Equation 8, the expected total regret satisfies the following inequality*

$$\begin{aligned}\mathbb{E}[R(T)] \leq \tau + \frac{7}{T} + L_\mu p\sqrt{\tau T}\left[2\sqrt{\frac{c_3}{\alpha}}g_1(J) + 2g_2(J)\right]\\ + L_\mu p\sqrt{\tau T}u\sqrt{\frac{2c_3 g_1(J)^2}{\alpha} + 2g_2(J)^2}\sqrt{1 + \log\lfloor\frac{T}{\tau}\rfloor},\end{aligned}$$

where $u = \sqrt{2\log(K\tau T^2)}$, $p = 1 + \frac{2}{\frac{1}{4\sqrt{\pi e}} - \frac{1}{T^2}}$ and $J = \lfloor\frac{T}{\tau}\rfloor$.

Remark 2. *Combining the choices of τ, α and the definition of $g_1(J), g_2(J)$ in Equation 6, 7, we have $\mathbb{E}[R(T)] \sim \tilde{O}(\sqrt{T})$. To study the dependence of regret bounds on d, we use a common condition in the literature (e.g., Li et al. (2017)) that $\sigma_0^2 \sim O(1)$ and make a similar assumption that $\lambda_f \sim O(1)$. As pointed out by the reader, this is unrealistic and a more proper assumption should be $\sigma_0^2, \lambda_f \sim O(1/d)$. We will discuss more about the dependencies on d in Section 8.8 in Appendix. In addition to the $\tilde{O}(\sqrt{T})$ theoretical guarantee of regret upper bound, our algorithm significantly improves efficiency when either T or d is large for GLB. To the best of our knowledge, it is by far the most efficient algorithm for GLB. See Table 1 in Section 6 for the comparisons of time complexity with other algorithms.[3] Moreover, the memory cost for UCB-GLM, GLM-TSL, SupCB-GLM and ϵ-greedy algorithms is linear in the total time horizon T, which could be very large in practice. For our proposed algorithm SGD-TS, the memory cost is a constant with respect to T.*

[3] Sherman–Morrison formula improves the time complexity of a matrix inverse in UCB-GLM and GLOC to $O(d^2)$.

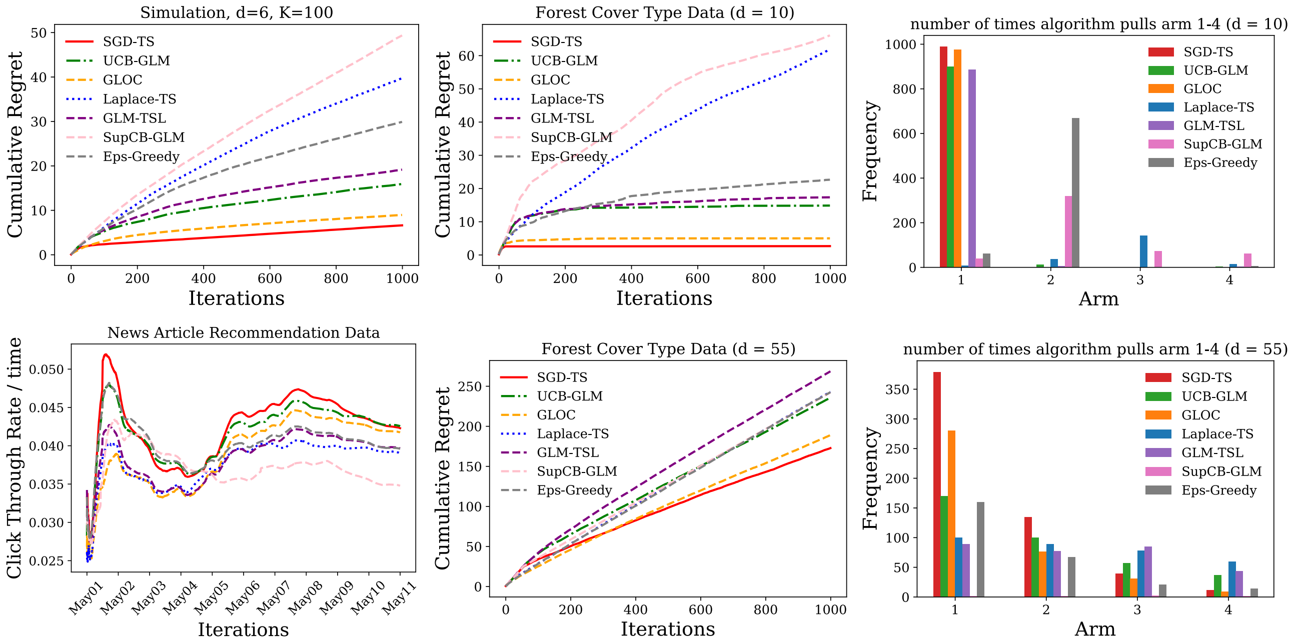

Figure 2: For the plots in the first two columns, from left to right, top to bottom, they are for simulation ($d = 6, K = 100$), scenario 1 for forest cover type data, news article recommendation data, scenario 2 for forest cover type data respectively. For the plots in the third column, they are the plots of the median frequencies of an algorithm pulls the best 4 arms for scenario 1 and 2 for forest cover type data respectively. (To reduce clutter, the legend in news article recommendation plot is omitted.)

6 EXPERIMENTAL RESULTS

In this section, we show by experiments in both synthetic and real datasets that our proposed SGD-TS algorithm outperforms existing approaches. We compare SGD-TS with UCB-GLM (Li et al., 2017), Laplace-TS (Chapelle and Li, 2011), GLOC (Jun et al., 2017), GLM-TSL (Kveton et al., 2020), SupCB-GLM (Li et al., 2017) and ϵ-greedy (Auer et al., 2002; Sutton et al., 1998).[4] In order to have a fair comparison, we perform a grid search for the parameters of different algorithms and select the best parameters to report. The covariance matrix in Equation 5 is set to $A_j = \frac{2a_1^2+2a_2^2}{j} I_d$, where a_1 and a_2 are explorations rates. We do a grid search for exploration rates of SGD-TS, GLOC, GLM-TSL, SupCB-GLM and UCB-GLM in $\{0.01, 0.1, 1, 5, 10\}$. The exploration probability of ϵ-greedy algorithm is set to $\frac{a}{\sqrt{t}}$ at round t and a is also tuned in $\{0.01, 0.1, 1, 5, 10\}$. As suggested by Li et al. (2017), τ should also be treated as a tuning parameter. For UCB-GLM, GLM-TSL, SupCB-GLM and SGD-TS, we set $\tau = \lfloor C \times \max(\log T, d) \rfloor$ and C is tuned in $\{1, 2, \dots, 10\}$. The initial step sizes η for SGD-TS, GLOC and Laplace-TS are tuned in $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. In SGD-TS, we set $\eta_j = \frac{\eta}{j}$. The experiments are repeated for 10 times and the averaged results are presented.

6.1 Simulation

We simulate a dataset with $T = 1000$, $K = 100$ and $d = 6$. The feature vectors and the true model parameter are drawn IID from uniform distribution in the interval of $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$. We build a logistic model on the dataset and draw random rewards Y_t from a Bernoulli distribution with mean $\mu(X_t^T \theta^*)$. As suggested by Dumitrascu et al. (2018), Laplace approximation of the global optimum does not always converge in non-asymptotic settings. Jun et al. (2017) points out that SupCB-GLM is an impractical algorithm. From Figure 2, we can see that our proposed SGD-TS performs the best, while SupCB-GLM and Laplace-TS perform poorly as expected.

6.2 News article recommendation data

We compare the algorithms on the benchmark Yahoo! Today Module dataset. This dataset contains 45,811,883 user visits to the news articles website -

[4] We choose UCB-GLM and GLOC since they have lower theoretical regrets than GLM-UCB and GLOC-TS respectively. We choose GLM-TSL over GLM-FPL since it was shown by Kveton et al. (2020) that GLM-TSL enjoys lower regret in practice.

Table 1: Comparison of time complexity and averaged runtime with other algorithms. The time complexity listed here assumes that T is much bigger than d. GLOC and Laplace-TS (only works for logistic bandit) need to solve an optimization problem on one datapoint every round and we assume this optimization problem can be solved in fixed iterations every round.

Algorithms	Time Complexity	Simulation	Yahoo
UCB-GLM (Li et al., 2017)	$O(T^2d)$	2.024	29.643
Laplace-TS (only for logistic bandit) (Chapelle and Li, 2011)	$O(Td)$	0.964	27.786
GLOC (Jun et al., 2017)	$O(Td^2)$	0.330	0.351
GLM-TSL (Kveton et al., 2020)	$O(T^2d^2)$	8.253	81.580
SupCB-GLM (Li et al., 2017)	$O(T^2d)$	4.609	26.842
ϵ-greedy (Auer et al., 2002; Sutton et al., 1998)	$O(T^2d)$	2.020	35.901
SGD-TS (This work)	$\mathbf{O(Td)}$	**0.099**	**0.127**

Yahoo Today Module from May 1, 2009 to May 10, 2009. For each user's visit, the module will select one article from a changing pool of around 20 articles to present to the user. The user will decide to click (reward $Y_t = 1$) or not to click ($Y_t = 0$). Both the users and the articles are associated with a 6-dimensional feature vector (including a constant feature), constructed by conjoint analysis with a bilinear model (Chu et al., 2009). We treat the articles as arms and discard the users' features. The click through rate (CTR) of each article at every round is calculated using the average of recorded rewards at that round. We still build logistic bandit on this data. Each time, when the algorithm pulls an article, the observed reward Y_t is simulated from a Bernoulli distribution with mean equal to its CTR. For better visualization, we plot $\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}[Y_s]$ against t. Since we want higher CTR, the result will be better if $\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}[Y_s]$ is bigger. From the plot in Figure 2, we can see that SGD-TS performs better than UCB-GLM during May 1 - May 2 and May 5 - May 9. During other days, UCB-GLM and SGD-TS have similar behaviors. However, other algorithms perform poorly in this real application.

6.3 Forest cover type data

We compare the algorithms on the Forest Cover Type data from the UCI repository. The dataset contains $581,021$ datapoints from a forest area. The labels represent the main species of the cover type. For each datapoint, if it belongs to the first class (Spruce/Fir species), we set the reward of this datapoint to 1, otherwise, we set it as 0. We extract the features (quantitative features are centralized and standardized) from the dataset and then partition the data into $K = 32$ clusters (arms). The reward of each cluster is set to the proportion of datapoints having reward equal to 1 in that cluster. Since the observed reward is either 0 or 1, we build logistic bandits for this dataset. Assume arm 1 has the highest reward and arm 4 has the 4-th highest reward. We plot the averaged cumulative regret and the median frequencies of an algorithm pulls the best 4 arms for the following two scenarios in Figure 2.

Scenario 1: Similar to Filippi et al. (2010), we use only the 10 quantitative features and treat the cluster centroid as the feature vector of the cluster. The maximum reward of the 32 arms is around 0.575 and the minimum is around 0.005.

Scenario 2: To make the classification task more challenging, we utilize both categorical and quantitative features, i.e., $d = 55$. Meanwhile, the feature vector of each cluster at each round is a random sample from that cluster. This makes the features more dynamic and the algorithm needs to do more exploration before being able to identify the optimal arm. The maximum reward is around 0.770 and the minimum is 0.

From the plots, we can see that in both scenarios, our proposed algorithm performs the best and it pulls the best arm most frequently. For scenario 1, GLOC, UCB-GLM and GLM-TSL perform relatively well, while the other algorithms are stuck in sub-optimal arms. This is consistent with the results in Dumitrascu et al. (2018). For the more difficult scenario 2, SGD-TS is still the best algorithm. GLOC performs relatively well, but it is not able to pull the best arm as frequently as SGD-TS. All the other algorithms perform poorly and frequently pull sub-optimal arms.

6.4 Computational cost

We present the averaged runtime of each algorithm for the simulation and Yahoo news article recommendation in Table 1. Presented results are the averaged runtime of one repeated experiment for one parameter combination in the grid search set. Note that all algorithms need to solve an optimization problem or invert a matrix each round except our algorithm. For example, UCB-GLM, GLM-TSL, SupCB-GLM and ϵ-greedy need to find MLE every round. Laplace-TS and GLOC need to solve an optimization problem on one data point every round. UCB-GLM, GLM-TSL, SupCB-GLM and GLOC need to calculate matrix inverse every round. For our proposed SGD-TS, since we only perform a single-step SGD update every round and do not need to calculate matrix inverse, so the real computational cost is the cheapest.

7 CONCLUSION AND FUTURE WORK

In this paper, we derive and analyze SGD-TS, a novel and efficient algorithm for generalized linear bandit. The time complexity of SGD-TS scales linearly in both total number of rounds and feature dimensions in general. Under the "diversity" assumption, we prove a regret upper bound of order $\widetilde{O}(\sqrt{T})$ for SGD-TS algorithm in generalized linear bandit problems. Experimental results of both synthetic and real datasets show that SGD-TS consistently outperforms other state-of-the-art algorithms. To the best of our knowledge, this is the first attempt that successfully applies online stochastic gradient descent steps to contextual bandit problems with theoretical guarantee. Our proposed algorithm is also the most efficient algorithm for generalized linear bandit so far.

Future work Although generalized linear bandit is successful in many cases, there are many other models that are more powerful in representation for contextual bandit. This motivates a number of works for contextual bandit with complex reward models (Chowdhury and Gopalan, 2017; Riquelme et al., 2018; Zhou et al., 2019). For most of these works, finding the posterior or upper confidence bound remains an expensive task in online learning. While we have seen in this work that online SGD can be successfully applied to GLB under certain assumptions, it is interesting to investigate whether we could further use online SGD to design efficient and theoretically solid methods for contextual bandit with more complex reward models, like neural networks, etc.

Acknowledgements

This research was partially supported by NSF HDR TRIPODS grant CCF-1934568, NSF IIS-1901527 and NSF IIS-2008173. JS is partially supported by NSF DMS 1712996. QD wants to thank Yi-Wei Liu for his useful comments. The authors also want to thank Kwang-Sung Jun for his helpful discussions.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Marc Abeille, Alessandro Lazaric, et al. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197, 2017.

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1948.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1, 2012.

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

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

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *Management Science*, 2020.

Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. *arXiv preprint arXiv:1802.04064*, 2018.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

Kani Chen, Inchi Hu, Zhiliang Ying, et al. Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs. *The Annals of Statistics*, 27(4):1155–1163, 1999.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 844–853. JMLR. org, 2017.

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, 2009.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In

Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 208–214, 2011.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

Shi Dong, Tengyu Ma, and Benjamin Van Roy. On the performance of thompson sampling on logistic bandits. In *COLT*, pages 1158–1160, 2019.

Bianca Dumitrascu, Karen Feng, and Barbara Engelhardt. Pg-ts: Improved thompson sampling for logistic contextual bandits. In *Advances in neural information processing systems*, pages 4624–4633, 2018.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems*, pages 99–109, 2017.

Sampath Kannan, Jamie H Morgenstern, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. In *Advances in Neural Information Processing Systems*, pages 2227–2236, 2018.

Branislav Kveton, Csaba Szepesvári, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic linear bandits. In *UAI*, page 176, 2019.

Branislav Kveton, Manzil Zaheer, Csaba Szepesvari, Lihong Li, Mohammad Ghavamzadeh, and Craig Boutilier. Randomized exploration in generalized linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 2066–2076, 2020.

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, 2010.

Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, pages 19–36, 2012.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2071–2080. JMLR. org, 2017.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *ICLR 2018 : International Conference on Learning Representations 2018*, 2018.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2014.

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

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

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.

Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.

Weiqiang Wu, Jing Yang, and Cong Shen. Stochastic linear contextual bandits with diverse contexts. *arXiv preprint arXiv:2003.02681*, 2020.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with upper confidence bound-based exploration. *arXiv preprint arXiv:1911.04462*, 2019.