Fast Generalized Stochastic Linear Bandit

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a generalized stochastic linear bandit problem and propose an algorithm that enjoys fast update. The computational complexity of the update is $\mathcal{O}(d)$, where d is the dimension of a context space. In comparison with other stochastic linear bandit algorithms, our algorithm does not need to incrementally update the inverse of a matrix so that it can avoid the $\mathcal{O}(d^2)$ computations. Yet, our problem has different assumptions. The reward that the learner receives in each round is binary and is generated by the logit model. We give some theoretical analysis of the proposed algorithm for the regret upper bound. Furthermore, we consider a distributed bandit setting such that there are some learners conducting the online learning. We extend our algorithm to the distributed setting. By communication, the learners can achieve speedup in learning, which is measured by regret. We also conduct experiments on two recommendation datasets (MovieLens and Yahoo! Front Page) to show that our algorithm not only updates faster but also achieves highly competitive click-through rate with the baseline.

1 Introduction

Online learning algorithms have drawn growing interests because of their theoretical guarantees and their applications in sequential learning and decision making. Among these algorithms, stochastic linear bandit (e.g. [3, 7, 13, 1]) has been adopted in web recommendation systems and shown to have some success. In the setting, in each round, the learner first observes some contexts (feature vectors) associated with some items. The learner then selects an item according to her decision rule. After that, the reward of choosing the item is revealed, while the rewards for choosing the others remain hidden. Finally, the learner updates her model based on her previous decisions and observations and then continues to the next round. The setting is so general that the related algorithms for the bandit problem have been applied to web advertising and recommendation [13, 14, 1]. However, previous works for stochastic linear bandit has a computational issue. Their algorithms update slowly when operating on a high dimensional context space. The reason is that the algorithms have to maintain the inverse of a matrix and the online update of the inverse matrix requires $\mathcal{O}(d^2)$ computations by Sherman–Morrison formula. As the consequence, the update is the bottleneck when the computational complexity of selecting an item is $o(d^2)$, which is usually the case in practice. This prevents from using rich feature representation of items. Thus, the performance for choosing good items is sacrificed for reducing response/update time and saving the computations.

In this work, we address this issue and propose an efficient algorithm; the complexity of update in the proposed algorithm scales linearly with the dimension of a context space. We achieve this by considering an assumption of reward which is different from the one in the original stochastic linear bandit problem. Previous works (e.g. [3, 7, 13, 1]) assume the expected reward of making a decision is the inner product of a unknown vector and the feature vector that corresponds to the decision. Instead, we study a generalized stochastic linear bandit problem which assumes a reward is generated by the logit model. The reward is binary and the probabilities of the outcomes are determined by the inner product of a unknown vector and the context vector associated with the decision through the logistic function. Since we consider the different assumption, one may challenge us that we do not really solve the computational issue of the original stochastic linear bandit problem. However, we argue that a different assumption of rewards is admissible as long as the new modeling and proposed algorithm has theoretical guarantee and can be competitive with or outperform the previous works in some applications such as a recommendation system.

We also extend our algorithm to a distributed bandit setting, which is another major contribution in this paper. The setting is that there are m learners doing online learning. The learners can communicate with a master to exchange their information. By communication, the learners can improve their learning and predicting performance. Our proposed algorithm allows the learners to achieve $\tilde{\mathcal{O}}(\sqrt{T/m})$ regret under certain condition, compared to $\tilde{\mathcal{O}}(\sqrt{T})$ when learning alone, where T is the number of rounds. We believe that the proposed distributed algorithm is useful for a recommendation system, as such a system usually needs to provide service to many users online at the same time.

Our results. We propose an efficient algorithm that enjoys fast update for a generalized stochastic linear bandit problem. We prove that the algorithm has $\mathcal{O}(\sqrt{T \log T})$ regret. Most importantly, the dimension of the context space does not appear explicitly in the regret bound. We also study the distributed bandit setting and propose a distributed algorithm that allows the learner to achieve speedup in learning. We conduct experiments and compare our algorithm with a popular stochastic linear bandit algorithm, which shows that our algorithm not only runs faster than the baseline but also achieves highly competitive click-through rate (CTR) with the baseline.

2 Preliminaries

As discussed in the introduction, we are interested in a specific stochastic linear bandit problem, which has recently been considered in [17]. In each round, the learner first makes a decision $\mathbf{x}_t \in \mathbb{R}^d$ from a decision set $\mathcal{D} \in \mathbb{R}^d$. Then, she receives a reward $r_t \in \mathbb{R}$. The reward is assumed to be binary $r_t \in \{0, 1\}$ and is generated from the logit model.

$$Pr[r_t = 1|\mathbf{x}_t] = \frac{1}{1 + \exp(-\mathbf{x}_t^\top \mathbf{w}^*)} = \frac{\exp(\mathbf{x}_t^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*)}, \tag{1}$$

and $Pr[r_t = 0|\mathbf{x}_t] = 1 - Pr[r_t = 1|\mathbf{x}_t]$, where $\mathbf{w}^* \in \mathbb{R}^d$ is a unknown vector. If $\mathbf{x}_t^\top \mathbf{w}^*$ is large, then the probability that observing the reward 1 is high, as the logistic function is monotone increasing with respect to the parameter $\mathbf{x}_t^\top \mathbf{w}^*$. This is why the problem is called generalized linear bandit, though the logistic function is nonlinear. The assumption of rewards can model the click ($r_t = 1$) or no-click ($r_t = 0$) of an advertisement ($\mathbf{x}_t$) in web advertising. For web advertising, one would like to design an algorithm to maximize the number of user clicks over time. Due to the assumption of rewards, the expected number of clicks achieved by an algorithm would be $\Sigma_{t=1}^T \exp(\mathbf{x}_t^\top \mathbf{w}^*)/(1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*))$. As our problem belongs to online learning, a common way to measure the performance of the learner is to compare the expected clicks she gets with the one by a clairvoyant who knows $\mathbf{w}^*$ in hindsight. The difference, which is called the *regret* of the learner, is

$$T \max_{x \in \mathcal{D}} \frac{\exp(\mathbf{x}^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}^\top \mathbf{w}^*)} - \sum_{t=1}^{T} \frac{\exp(\mathbf{x}_t^\top \mathbf{w}^*)}{1 + \exp(\mathbf{x}_t^\top \mathbf{w}^*)}. \tag{2}$$

However, we do not analyze the regret bound of an algorithm based on the definition above due to some technical difficulties. Instead, we provide an upper bound of the following measure,

$$T \max_{x \in \mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^*, \tag{3}$$

while we still use the assumption of the rewards when deriving the upper bound. One can show that (2) and (3) are at the same order. Denote the value of (2) as (A) and the value of (3) as (B). [17] has shown that $\frac{1}{2(1+\exp(\theta))}$ (B) $\leq$ (A) $\leq \frac{1}{4}$ (B) , assuming $\|(\mathbf{w}^*)^\top \mathbf{x}\|_2 \leq \theta$ for any $\mathbf{x} \in \mathcal{D}$. Consequently, the derived upper bound of (3) is within a constant multiple of (2).

Related works. There exist some works considering the generalized stochastic linear bandit. [8] studies the case that the rewards of making a decision $\mathbf{x}_t \in \mathcal{D}$ in round t satisfy $\mathbb{E}[r_t] = \mu(\mathbf{x}_t^\top \mathbf{w}^*)$, where function μ satisfies some main properties of the generalized linear model in statistics literature. The logistic function $\mu(x) = \exp(x)/(1 + \exp(x))$ can be the one. However, their proposed algorithm needs to save the history of decisions and observations, which means that the computational complexity and space usage in round t is $\mathcal{O}(t)$. [17] considers the same assumption of rewards (1) and the same regret definition (2). The goal for providing the upper bound of (3) instead of (2) in this paper follows theirs. Yet, our algorithm is different so that the theoretical analysis is not the same. Furthermore, their algorithm uses online newton method ([9]). As a result, the update still involves maintaining the inverse of a matrix, which is $\mathcal{O}(d^2)$. In addition, the regret bound in [17] is worse than ours by a factor of $\tilde{\mathcal{O}}(d)$, as we will see.

Most of the works for stochastic linear bandit (e.g. [3, 7, 13, 1]) consider $\mathbb{E}[r_t] = \mathbf{x}_t^\top \mathbf{w}^*$. They leverage the idea of online least squares regression so that incrementally maintaining the inverse of a matrix seems to be unavoidable. For this computational issue, [10] proposes an algorithm whose update is like online gradient descent. However, they do not provide theoretical analysis for the algorithm. Besides, the algorithm does not perform well as compared to the popular one [1] on Yahoo! Front Page dataset; its CTR is only 75% of the one by [1]. In comparison, our algorithm achieves highly competitive CTR score with [1]. We also provide theoretical guarantees for our algorithm.

For distributed bandit, [4] considers the case of distributed stochastic bandit, which is a distributed version of stochastic multi-arm bandit [3]. It assumes a peer-to-peer communication environment, which means that each learner can communicates with each other by sending messages along the links of a overlay network. Very recently, [11] extends the setting to distributed stochastic linear bandit. Yet, the assumption of rewards in [11] follows [1], while in our work, it is based on [17]. Furthermore, we assume a different communication protocol. In our work, it is a master's job to communicates with the learners, and the learners do not communicate with each other directly. The master/slaves model is suitably for many modern computer architectures that leverage distributed computing resources. For example, a server creates multiple threads or processes to provide recommendations to some users at the same time.

3 Algorithm

Let us begin by giving another assumption and notation. Without loss of generality, we assume for every $\mathbf{x} \in \mathcal{D}$ in the decision space, its L2 norm satisfies

$$l \le \|\mathbf{x}\|_2 \le u. \tag{4}$$

We also assume the unknown vector $\mathbf{w}^* \in \mathbb{R}^d$ is in a L2 norm ball whose radius is R, $\|\mathbf{w}^*\|_2 \le R$. Moreover, for brevity, the probability of the binary reward $r_t = 0$ is mapped to $r_t = -1$ now, so the probability (1) can be written into a compact form, i.e. $Pr[r_t = \{\pm 1\}|\mathbf{x}_t] = \frac{1}{1+\exp(-r_t \mathbf{x}_t^\top \mathbf{w}^*)} = \frac{\exp(r_t \mathbf{x}_t^\top \mathbf{w}^*)}{1+\exp(r_t \mathbf{x}_t^\top \mathbf{w}^*)}$. Let us denote $f_t(\mathbf{w}) = \log(1 + \exp(-r_t \mathbf{x}_t^\top \mathbf{w}))$, which is the logistic loss function. It follows that maximizing the probability function $\frac{1}{1+\exp(-r_t \mathbf{x}_t^\top \mathbf{w})}$ over $\mathbf{w}$ is equal to minimizing the logistic loss $f_t(\mathbf{w})$.

Our algorithm is shown in the following block, where Π is the projection into the ball, and $\nabla f_t(\mathbf{w}_t)$ is the gradient of the function $f_t(\mathbf{w}) = \log(1 + \exp(-r_t \mathbf{x}_t^\top \mathbf{w}))$ at point $\mathbf{w}_t$. The algorithm requires two parameters, which are the radius of a confidence ball γ_t and the learning rate η_t. Both parameters are defined in the following section.

Algorithm 1 Our algorithm

```
Require $\gamma_t$ and $\eta_t$.
Initialization: Let $\mathbf{w_1} = \mathbf{0} \in \mathbb{R}^d$
    $(\mathbf{x}_t, \hat{\mathbf{w}}_t) = \arg\max_{\mathbf{x} \in \mathcal{D}, \mathbf{w} \in \mathcal{C}_t} \mathbf{x}^\top \mathbf{w}$, where $C_t = \{\mathbf{w} : \|\mathbf{w} - \mathbf{w}_t\|_2 \le \sqrt{\gamma_t}\}$.
    Select $\mathbf{x}_t$ and observe reward $r_t = 1$ or $r_t = -1$ .
    Update $\mathbf{w}_{t+1} = \Pi(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t))$
```

Clearly, the update (line 3 in Algorithm 1) is $\mathcal{O}(d)$. For the optimization problem in line 1, if the decision set $\mathcal{D}$ is finite: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|\mathcal{D}|}\}$, then $\mathbf{x}_t = \arg\max_{x_k} \mathbf{w}_t^\top \mathbf{x}_k + \sqrt{\gamma_t}\|\mathbf{x}_k\|_2$. If the decision space $\mathcal{D}$ is infinite but is the unit ball, then the problem still admits being solved efficiently.

4 Theoretical analysis

In this section, we analyze the regret of Algorithm 1. As mentioned in the preliminaries section, our goal is to provide the upper bound of (3) for our algorithms.

We first give some lemmas that are used to construct Theorem 1.

Lemma 1. *([17]) Assume* $\mathbf{w} \in \{\mathbf{w} : \|\mathbf{w}\|_2 \leq R\}$*, the following holds for* $\beta \leq \frac{1}{2(1+\exp(R))}$*:* $f_t(\mathbf{w}_2) \geq f_t(\mathbf{w}_1) + \nabla f_t(\mathbf{w}_1)^\top(\mathbf{w}_2 - \mathbf{w}_1) + \frac{\beta}{2}((\mathbf{w}_2 - \mathbf{w}_1)^\top \mathbf{x}_t)^2$.

Next, for each function $f_t(\cdot)$, let $\bar{f}_t(\cdot) = \mathbb{E}_{y_t}[\log(1 + \exp(-y_t \mathbf{x}_t^\top \mathbf{w})]$ be its conditional expectation over y_t. Then, we have

Lemma 2. *([17])* $\bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) = D_{KL}(p_{\mathbf{w}^*}||p_{\mathbf{w}}) \geq 0$*, where* $D_{KL}(\cdot||\cdot)$ *means the KL-divergence.*

The following theorem shows that with high probability, the distance between our algorithm's $\mathbf{w}_t$ and the unknown $\mathbf{w}^*$ in each round t is scaled with $\mathcal{O}(1/t)$.

Theorem 1. *Let* $\delta \in (0, 1/e)$ *and assume* $T \geq 4$*. If* $\eta_t = \frac{2}{\beta l t}$*, then with probability at least* $1 - \delta$*, we have* $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{624u^2 \log(T\log(T)/\delta) + 4u^2}{\beta^2 l^2 t}$*, for any* $t \leq T$*, where* $\beta \leq \frac{1}{2(1+\exp(R))}$*,* l *and* u *are the lower and upper bound of the L2 norm of any* $\mathbf{x} \in \mathcal{D}$ *respectively.*

Proof. (sketch) $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\Pi_R(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)) - \mathbf{w}^*\|^2 \leq \|\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t) - \mathbf{w}^*\|^2$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t\langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t\langle \nabla \bar{f}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$\leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t(\bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) + \frac{\beta l}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2) + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t(D_{KL}(p_{\mathbf{w}^*}||p_{\mathbf{w}}) + \frac{\beta l}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2) + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$\leq (1 - \eta_t \beta l)\|\mathbf{w}_t - \mathbf{w}^*\|^2 + 2\eta_t\langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 u^2$

In the above, the first inequality is the known property of projection. The second inequality is due to Lemma 1. Taking the conditional expectation on Lemma 1 and rearranging it leads $\nabla \bar{f}_t(\mathbf{w}_t)^\top(\mathbf{w}_t - \mathbf{w}^*) \geq \bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) + \frac{\beta}{2}((\mathbf{w}_t - \mathbf{w}^*)^\top \mathbf{x}_t)^2$. By combining the lower bound assumption of $\|\mathbf{x}\|_2$, we have the inequality. The subsequent equality is due to Lemma 2. The last inequality is because $\|\nabla f_t(\mathbf{w}_t)\|^2 \leq u^2$ and the fact that KL-divergence is positive.

Unwinding the derived inequality till $t = 2$, we get $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{2}{\beta l}\Sigma_{i=2}^t \frac{1}{i}(\Pi_{j=i+1}^t(1 - \frac{2}{t}))\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^*\rangle + \frac{u^2}{\beta^2 l^2}\Sigma_{i=2}^t \frac{1}{i^2}(\Pi_{j=i+1}^t(1 - \frac{2}{t}))$. With further simplification, one can show that

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{2}{\beta l t(t-1)}\Sigma_{i=2}^t (i-1)\langle \nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^*\rangle + \frac{u^2}{\beta^2 l^2 t}. \tag{5}$$

We can continue to provide the bound of the distance. The process is similar as the proof for Proposition 1 in [15]. For the details, please see Appendix A of the supplementary.

□

Theorem 2. *Set* $\gamma_t = \frac{624u^2 \log(T\log(T)/\delta) + 4u^2}{\beta^2 l^2 t}$ *and* $\eta_t = \frac{2}{\beta l t}$ *in Algorithm 1. Let* $\delta \in (0, 1/e)$ *and assume* $T \geq 4$*. Then, with probability at least* $1 - \delta$*, we have* $T\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* \leq 2\sqrt{\frac{624u^4 \log(T\log(T)/\delta) + 4u^4}{\beta^2 l^2}(\log(T) + 1)T}$.

Proof. Let $\mathbf{x}^*$ be the optimum of $\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^*$. Then $T\max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* = \Sigma_{t=1}^T \mathbf{x}^{*\top}\mathbf{w}^* - \mathbf{x}_t^\top \mathbf{w}^* \leq \Sigma_{t=1}^T \mathbf{x}_t^\top \hat{\mathbf{w}}_t - \mathbf{x}_t^\top \mathbf{w}^* = \Sigma_{t=1}^T \mathbf{x}_t^\top(\hat{\mathbf{w}}_t - \mathbf{w}_t) + \mathbf{x}_t^\top(\mathbf{w}_t - \mathbf{w}^*) \leq$

$\Sigma_{t=1}^{T}\|\mathbf{x}_t\|_2(\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|_2 + \|\mathbf{w}_t - \mathbf{w}^*\|_2) \leq \Sigma_{t=1}^{T}\|\mathbf{x}_t\|_2(\sqrt{\gamma_t} + \sqrt{\gamma_t})$, where the first inequality is due to the optimization problem in line 1 in Algorithm 1. The inequality $\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|_2 \leq \sqrt{\gamma_t}$ holds because of the constraint in line 1 of the algorithm. Moreover, the term $\|\mathbf{w}_t - \mathbf{w}^*\|_2$ is bounded by $\sqrt{\gamma_t}$ for all t with probability $1-\delta$, according to Theorem 1.

Thus, we have $T\max_{x\in\mathcal{D}}\mathbf{x}^\top\mathbf{w}^* - \Sigma_{t=1}^{T}\mathbf{x}_t^\top\mathbf{w}^* \leq 2\Sigma_{t=1}^{T}\sqrt{\gamma_t}\|\mathbf{x}_t\|_2 \leq 2\sqrt{\Sigma_{t=1}^{T}\gamma_t}\sqrt{\Sigma_{t=1}^{T}\|\mathbf{x}_t\|_2^2} \leq 2\sqrt{u^2T}\sqrt{\Sigma_{t=1}^{T}\gamma_t} \leq 2\sqrt{u^2T}\sqrt{\Sigma_{t=1}^{T}\frac{624u^2\log(T\log(T)/\delta)+4u^2}{\beta^2l^2t}}$ $\leq 2\sqrt{\frac{624u^4\log(T\log(T)/\delta)+4u^4}{\beta^2l^2}(\log(T)+1)T}$, where the second inequality is by using Cauchy-Schwarz inequality. □

Theorem 2 means that Algorithm 1 can achieve $\mathcal{O}(\sqrt{T\log T})$ regret of (2). The upper bound does not depend explicitly on the dimension of the context space, d. The dimension is implicitly connected to the bound through the L2 norm assumption of the decision set, namely, (4). This regret upper bound is $\mathcal{O}(d)$ improvement over the one by [17]. The computational complexity is also $\mathcal{O}(d)$ improvement over [17]. We think that the implication is significant. All the related works for stochastic linear bandit [1, 3, 5, 7, 8, 13, 16] have d or $\sqrt{d}$ factor in their regret upper bounds, which means that in additional to the computational issue, the performance in terms of receiving a reward such as a click on an advertisement, would degrade in a high dimensional contextual space. As the consequence, those previous works may not admit rich feature representation that may improve the performance in applications.

The above theoretical analysis assumes the decision space $\mathcal{D}$ is fixed in each round. Yet, the assumption is not necessary. If the decision space is changed over rounds ($\mathcal{D}_t$ instead of $\mathcal{D}$), the analysis can still proceed. We can also extend the algorithm to a case that the learner can recommend more than one item in each round. To be specific, assume the decision set is finite, $\mathcal{D}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \cdots, \mathbf{x}_{|\mathcal{D}_t|,t}\}$, where each $\mathbf{x}_{i,t}$ is the feature representation of an item. The learner can pick k items from $\mathcal{D}_t$ in each round, where $k \geq 1$. Then, the goal of the learner should be choosing the best subset of k items that maximizes the number of clicks. In the supplementary, we provide a variant of Algorithm 1 that achieves $\mathcal{O}(k\sqrt{T\log T})$ regret in this combinatorial setting.

5 Distributed generalized stochastic linear bandit

In this section, we extend our algorithm to a distributed scenario. The motivation is that in a typical recommendation system, there are many users interacting with the system at the same time. Specifically, we consider the scenario that there are m learners; each provides recommendation to a user at a time. By communication, the learners can improve their learning and predicting performance.

We assume the distributed architecture is that their exists a master communicating with the m learners. The master maintains and updates a global parameter, while a learner uses a global parameter to provide recommendation and receives the feedback from a user. Our assumption for the communication protocol adopts the cyclic delayed update fashion (round-robin fashion), which has been considered in [12, 2] in the distributed optimization literature.

Our distributed algorithm for the learners and the master are shown on Algorithm 2 and Algorithm 3 respectively. The master maintains a global index t. At each t, the master communicates with a learner in the cyclic fashion and exchange the information. It is the master that performs the update of the global parameter $\mathbf{w}$, while a learner makes a decision, receives the feedback, and computes the gradient. The master performs the updates by using out-of-date information (last line in Algorithm 3) due to the communication protocol. Because of the cyclic delayed protocol, the gradient used for the update at step t is computed from the global parameter obtained at step $t-m$. For step 1 to step m, one can execute Algorithm 1 to initialize the master and the learners.

In the following, we analyze our distributed algorithm. The proofs are available in the Appendix B.

Theorem 3. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$. If $\eta_t = \frac{1}{\beta l t}$, then with probability at least $1-\delta$, we have $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{624u^2\log(T\log(T)/\delta)+4u^2(1+\frac{u}{\beta l}m(\log m+2))}{\beta^2l^2t}$, for any $t \leq T$ of our distributed algorithm, where $\beta \leq \frac{1}{2(1+\exp(R))}$, l and u are the lower and upper bound of the L2 norm of any $\mathbf{x} \in \mathcal{D}$ respectively.*

Algorithm 2 Distributed algorithm (learner)

```
Receive $\mathbf{v}$ and $\theta$ from the master
$(\mathbf{x}, \hat{\mathbf{w}}) = \arg\max_{\mathbf{x}\in\mathcal{D}, \mathbf{w}\in\mathcal{C}} \mathbf{x}^\top \mathbf{w}$, where $C = \{\mathbf{w} : \|\mathbf{w} - \mathbf{v}\|_2 \leq \sqrt{\theta}\}$.
Select $\mathbf{x}$ and observe reward $r = 1$ or $r = -1$ .
Compute $\nabla g(\mathbf{v}) = \nabla \log(1 + \exp(-r\mathbf{x}^\top \mathbf{v}))$.
Send $\nabla g(\mathbf{v})$ when the master calls it again.
```

Algorithm 3 Distributed algorithm (master)

```
For $t = m + 1, \ldots, T$
    Communicate with a learner (in the cyclic fashion).
    Send $\mathbf{v} = \mathbf{w}_t$ and $\theta = \gamma_t$ to the learner
    Receive $\nabla f_t(\mathbf{w}_{t-m}) = \nabla g(\mathbf{w}_{t-m})$ from the learner.
    Update $\mathbf{w}_{t+1} = \Pi(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_{t-m}))$
```

Using Theorem 3, we can derive the regret of the distributed algorithm.

Theorem 4. *For our distributed algorithm, set* $\gamma_t = \frac{624u^2 \log(T\log(T)/\delta) + 4u^2(1 + \frac{u}{\beta l} m(\log m + 2))}{\beta^2 l^2 t}$ *and* $\eta_t = \frac{1}{\beta l t}$ *and let* $\delta \in (0, 1/e)$*. Assume* $T \geq 4$*. Then, with probability at least* $1 - \delta$*, we have* $T \max_{x\in\mathcal{D}} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{t=1}^T \mathbf{x}_t^\top \mathbf{w}^* \leq 2\sqrt{\frac{624u^4 \log(T\log(T)/\delta) + 4u^4(1 + \frac{u}{\beta l} m(\log m + 2))}{\beta^2 l^2}}(\log(T) + 1)T.$

Recall that T is the index maintained by the master, and it is the total number of rounds conducted by all the m learners. That is, each learner conducts T/m rounds, assuming T is a multiple of m. According to the theorem, if $154\log(T\log(T)/\delta) \gg 1 + \frac{u}{\beta l} m(\log m + 2)$, then the term introduced by the delay is negligible. The meaning is that each learner has $\mathcal{O}(\sqrt{m})$ speedup in learning compared to the case when learning alone; if each learner independently processes T/m rounds without communication, then the total regret would be $m\tilde{\mathcal{O}}(\sqrt{T/m}) = \tilde{\mathcal{O}}(\sqrt{Tm})$. The condition for $\mathcal{O}(\sqrt{m})$ speedup holds when the ratio u/l and m is not too large. On the other hand, if both quantities are at the same order, then the total regret would be $\tilde{\mathcal{O}}(\sqrt{Tm})$ regret which means no speedup is achieved, compared to the performance of learning without communication.

6 Experiment

In the experiments, we compare our algorithm (Algorithm 1) with [1], the popular stochastic linear bandit algorithm. The experiments are conducted on two datasets.

The first dataset is Yahoo! Webscope dataset (R6A) [1]. It is the benchmark of measuring and comparing performance of bandit algorithms [6, 14]. Each line in the log files represents a user interacting with one randomly chosen article from a pool of articles. It records a click ($r_t = 1$) or no-click ($r_t = 0$) for the recommended article. The articles available to present to a user in each round is the subset of the articles. That is, the decision set $\mathcal{D}_t$ may be different over time. Each record in the log file is obtained by randomly and uniformly selecting an available article for recommendation. The way of collecting the records can be used to construct an unbiased estimator of the performance for a bandit algorithm. [14] suggests an algorithm being evaluated to step through the log files line by line. If the algorithm recommends the same article as the one recorded in the line, the event is added to the history and the algorithm is updated; otherwise, it just simply ignore the line. The measure of the performance is CTR score, defined as the number of clicks divided by the number of retained events (records).

Each user and each article in the Yahoo! R6A dataset is represented by a 6 dimensional feature vector. Based on the raw feature vectors, we construct some rich feature representation for the available articles in each round. The way we form the high-dimensional features is described as follows. Denote a raw vector as $\mathbf{v} = [v[1], v[2], \ldots, v[6]]^\top$. We can generate the m_{th} order representation as $\mathbf{u} = [v[1], v[2], \ldots, v[6], v[1]^2, v[2]^2, \ldots, v[6]^2, \ldots, v[1]^m, v[2]^m \ldots, v[6]^m]^\top$. Then, the final

[1] `https://webscope.sandbox.yahoo.com/`

Table 1: Performance of the baseline on Yahoo! R6A dataset.

feature dimensions	baseline CTR	baseline average update time (s)	baseline average running time (s)
36 (order=1)	5.211 %	3.80×10^{-4}	5.95×10^{-4}
576 (order=4)	5.350 %	7.47×10^{-2}	9.07×10^{-2}
3600 (order=10)	n/a	2.88	3.40

Table 2: Performance of Algorithm 1 on Yahoo! R6A dataset.

feature dimensions	Algorithm 1 CTR	Algorithm 1 average update time (s)	Algorithm 1 average running time (s)
36 (order=1)	4.876 %	1.79×10^{-5}	1.36×10^{-4}
576 (order=4)	5.533 %	4.08×10^{-4}	7.00×10^{-3}
3600 (order=10)	5.187 %	2.90×10^{-3}	3.62×10^{-2}

constructed feature vectors of the articles are obtained by conducting the outer product of each m_{th} order representation of a user vector and available articles' vectors in each round. For the m_{th} order, it generates a $(m \times 6)^2$ dimensional feature vector for each article.

The second dataset is MovieLens 10M dataset [2]. This dataset consists of tuples (user's ID, movie's ID, rating score [1-5]). We assume that the ratings which are less than 4 as no-click ($r_t = 0$), the other cases are click ($r_t = 1$). We try to simulate the bandit problem as the first dataset. The way we construct a pseudo log file is as follows. First, we choose the 200 movies that get most clicks and the 200 movies that get most no-clicks. The union is the set of 324 movies. Then, a tuple in the original rating file is randomly sampled and at the same time the decision space is constructed by randomly sampling 25 items from the pool of 324 movies. If the movie indicated by the sampled tuple is in the sampled decision set, then the tuple and decision space is added to the pseudo log file. The procedure is repeated to construct about 300 thousands records in the pseudo log file. For the feature representation, we use LIBMF [3], a matrix factorization toolkit, to construct the items' features based on the ratings. The feature dimension is set to 100, 500, and 1000 in the experiment. The evaluation follows the same procedure as the Yahoo! R6A dataset.

There are parameters for the baseline and our algorithm. Denote the feature vector of an available item k in round t as $\mathbf{x}_{t,k}$. For the algorithm of [1], the score of each item when making a decision is computed as $\mathbf{w}_t^\top \mathbf{x}_{t,k} + \alpha_1 \sqrt{\mathbf{x}_{t,k}^\top \mathbf{M}_t^{-1} \mathbf{x}_{t,k}}$, where $\mathbf{w}_t$ is the online least squares solution and $\mathbf{M}_t$ is the matrix that facilitates exploration (please see [1] for details). We set α_1 as a tuning parameter. For our algorithm, there are two parameters η_t and γ_t. For η_t, we set $\eta_t = \frac{150}{t}$ for the MovieLens dataset, and set $\eta_t = \frac{\alpha_2}{t}$ for the Yahoo! R6A dataset, where α_2 is a tuning parameter. For γ_t, we set $\gamma_t = \frac{624 \log(T \log(T)/0.01)+4}{\beta^2 (\alpha_3 t)^2}$, where α_3 is a tuning parameter . In both algorithms, when making a decision, computing the score of each item is parallelizable (e.g. line 1 in Algorithm 1 in our case). That is, a number of threads can be created and each thread can compute the scores for some items at the same time. We use OpenMP/C++ to achieve that. The other computations such as updating the models in both algorithms do not exploit the parallel computing. Codes to reproduce the experiments are available in the supplementary.

6.1 Experiment results on Yahoo! Webscope dataset (R6A)

The algorithms are executed on the 05/01/2009 log file in the dataset, which consists of 4 million records. The original random policy for collecting the records achieves 3.10 % CTR. Table 1 and Table 2 show the performance of the baseline and our algorithm respectively. The third column and the fourth column in the tables represent average computing time in each round. Since the number of retained records (i.e. number of effective rounds) by each algorithm during evaluation is different, the average computing time in each round is reported. The total computing time is divided by the number of retained records to obtain the average. The update time represents the time for updating

[2] http://grouplens.org/datasets/movielens/

[3] https://www.csie.ntu.edu.tw/~cjlin/libmf/

Table 3: Performance of the baseline on MovieLens dataset.

feature dimensions	baseline CTR	baseline average update time (s)	baseline average running time (s)
100	0.8549	5.80×10^{-3}	1.46×10^{-2}
500	0.8640	1.54×10^{-1}	2.80×10^{-1}
1000	0.8579	1.67	2.08

Table 4: Performance of Algorithm 1 on MovieLens dataset.

feature dimensions	Algorithm 1 CTR	Algorithm 1 average update time (s)	Algorithm 1 average running time (s)
100	0.8657	5.36×10^{-5}	1.40×10^{-3}
500	0.8697	3.22×10^{-4}	4.40×10^{-3}
1000	0.8673	4.88×10^{-4}	1.44×10^{-2}

an algorithm's model; in our case, it is corresponds to the last line in Algorithm 1, while in the baseline, it is corresponds to updating the inverse matrix and obtaining the least-squares-regression-like solution. The running time consists of the update time as well as the time for making a decision.

As we can see from the tables, our algorithm is significantly faster than the baseline. Moreover, the CTR scores of our algorithm are highly competitive with the baseline. For the tenth order feature representation, the baseline cannot finish the experiment in three weeks, so we could not provide the CTR (n/a). The tables also show that there are some improvements in CTR using high dimensional feature vectors. The CTR scores of both algorithms are higher when using the fourth order features, compared to the ones using the first order features. Yet, when using a much higher feature vectors (i.e. order=10), the improvement is degraded to some degree. However, as the dimension of raw features of this dataset is 6, constructing high dimensional features may be hard.

6.2 Experiment results on MovieLens dataset

Table 3 and Table 4 show the performance of the baseline and our algorithm respectively. In this dataset, a policy that randomly and uniformly choose an item in each round has 0.65 CTR. From the tables, we see that our algorithm is better than the baseline measured by both CTR and running time.

6.3 Simulation for distributed bandit

Figure 1 shows the simulation results for our distributed algorithm on Yahoo! R6A dataset, which are the CTRs with respect to number of rounds per learner under different values of parameter m. We can see that the gain increases with number of learners m till some points (m=50). Yet, even for $m = 100$, the learning rate of each learner is much faster than the one of learning without communication ("no delay" in the figure). This demonstrates that our algorithm works well in the distributed setting.

7 Conclusion

In this paper, we propose an efficient algorithm whose update is fast and regret upper bound does not depend explicitly on the dimension of the context space. Our algorithm admits using high dimensional context vectors, which offers flexibility for feature engineering. The significance is reflected on the benchmark dataset. Furthermore, we develop a distributed algorithm and analyze its regret. We believe that it is a big step towards developing distributed bandit algorithms that work well in applications with theoretical guarantees. Future works include implementing our distributed algorithm in a real recommendation system as well as considering different communication protocols.

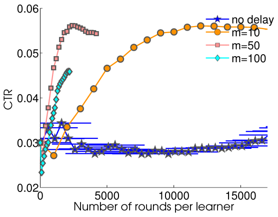


Figure 1: Simulation for distributed bandit.

References

[1] Y. Abbasi-yadkori, D. Pál, A. Garivier, and C. Szepesv́ari. Improved algorithms for linear stochastic bandits. *NIPS*, 2011.

[2] A. Agarwal and J. Duchi. Distributed delayed stochastic optimization. *NIPS*, 2011.

[3] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, page 3:397422, 2002.

[4] S. Balázs, Róbert B.-F., H. István, O. Róbert, J. Márk, and K. Balázs. Gossip-based distributed stochastic bandit algorithms. *ICML*, 2013.

[5] W. Chu, L. Li, L. Reyzin, and E. Schapire. Contextual bandits with linear payoff functions. *AISTATS*, 2011.

[6] W. Chu, S. Park, T. Beaupre, N. Motgi, Amit Phadke, and J. Chakraborty, S. Zachariah. A case study of behavior-driven conjoint analysis on yahoo!: front page today module. *KDD*, 2009.

[7] D. Dani, T. Hayes, and S. Kakade. Stochastic linear optimization under bandit feedback. *COLT*, 2008.

[8] S. Filippi, O. Cappe, A. Garivier, and C. Szepesv́ari. Parametric bandits: The generalized linear case. *NIPS*, 2011.

[9] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(203):169–192, 2007.

[10] N. Korda, L.A. Prashanth, and R. Munos. Fast gradient descent for drifting least squares regression, with application to bandits. *AAAI*, 2015.

[11] N. Korda, B. Szörényi, and S. Li. Distributed clustering of linear bandits in peer to peer networks. *ICML*, 2016.

[12] J. Langford, A. Smola, and M. Zinkevich. Slow learners are fast. *NIPS*, 2009.

[13] L. Li, W. Chu, J. Langford, and R. Schapire. A contextual-bandit approach to personalized news article recommendation. *WWW*, 2010.

[14] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *WSDM*, 2011.

[15] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *ICML*, 2012.

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

[17] L. Zhang, T. Yang, R. Jin, and Z. Zhou. Online stochastic linear optimization under one-bit feedback. *ICML*, 2016.

Fast Generalized Stochastic Linear Bandit (Supplementary Material)

Anonymous Author(s)
Affiliation
Address
`email`

We first give some lemmas that are used to construct our theorems.

Lemma 1. *([2]) Assume* $\mathbf{w} \in \{\mathbf{w} : \|\mathbf{w}\|_2 \le R\}$*, the following holds for* $\beta \le \frac{1}{2(1+\exp(R))}$*:* $f_t(\mathbf{w}_2) \ge f_t(\mathbf{w}_1) + \nabla f_t(\mathbf{w}_1)^\top(\mathbf{w}_2 - \mathbf{w}_1) + \frac{\beta}{2}((\mathbf{w}_2 - \mathbf{w}_1)^\top \mathbf{x}_t)^2$.

Lemma 2. *Assume* $\mathbf{w} \in \{\mathbf{w} : \|\mathbf{w}\|_2 \le R\}$*, the following holds for* $L \ge \frac{1}{4}$*:* $f_t(\mathbf{w}_2) \le f_t(\mathbf{w}_1) + \nabla f_t(\mathbf{w}_1)^\top(\mathbf{w}_2 - \mathbf{w}_1) + \frac{L}{2}((\mathbf{w}_2 - \mathbf{w}_1)^\top \mathbf{x}_t)^2$.

Next, for each function $f_t(\cdot)$, let $\bar{f}_t(\cdot) = \mathbb{E}_{y_t}[\log(1 + \exp(-y_t\mathbf{x}_t^\top \mathbf{w})]$ be its conditional expectation over y_t. Then, we have the following lemma which is due to [2].

Lemma 3. *([2])* $\bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) = D_{KL}(p_{\mathbf{w}^*}||p_{\mathbf{w}}) \ge 0$*, where* $D_{KL}(\cdot||\cdot)$ *means the KL-divergence.*

Proof. ([2]) Recall that the distribution of rewards for making a decision $\mathbf{x}_t$ follows $p_{\mathbf{w}^*}(r) = \frac{1}{1+\exp(-r\mathbf{x}_t^\top \mathbf{w}^*))}$, where $r = \pm 1$. Therefore, according to the definition of conditional expectation, we have $\bar{f}_t(\mathbf{w}) = p_{\mathbf{w}^*}(1)(-\log(p_{\mathbf{w}}(1))) + p_{\mathbf{w}^*}(-1)(-\log(p_{\mathbf{w}}(-1))) = -\Sigma_{y\in\{\pm1\}} p_{\mathbf{w}^*}(y)\log p_{\mathbf{w}}(y)$.

Thus, $\bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) = -\Sigma_{y\in\{\pm1\}} p_{\mathbf{w}^*}(y)\log p_{\mathbf{w}}(y) - \Sigma_{y\in\{\pm1\}} p_{\mathbf{w}^*}(y)\log p_{\mathbf{w}^*}(y) = \Sigma_{y\in\{\pm1\}} p_{\mathbf{w}^*}(y)\log \frac{p_{\mathbf{w}^*}(y)}{p_{\mathbf{w}}(y)} = D_{KL}(p_{\mathbf{w}^*}||p_{\mathbf{w}})$. □

Lemma 4. $\|\mathbf{w}_t - \mathbf{w}^*\|_2 \le \frac{2u}{\beta l}$.

Proof. We have $u\|(\mathbf{w}_t - \mathbf{w}^*)\|_2 \ge \|\mathbf{x}_t\|_2\|(\mathbf{w}_t - \mathbf{w}^*)\|_2 \ge \|\nabla\bar{f}_t(\mathbf{w}_t)\|_2\|(\mathbf{w}_t - \mathbf{w}^*)\|_2 \ge \nabla\bar{f}_t(\mathbf{w}_t)^\top(\mathbf{w}_t - \mathbf{w}^*) \ge \frac{\beta l}{2}\|\mathbf{w}_t - \mathbf{w}^*\|_2^2$. The first inequality is due to the assumption. The second inequality is because $\|\nabla f_t(\mathbf{w}_t)\|^2 = (\frac{\exp(-y_t\mathbf{x}_t^\top w)}{1+\exp(-y_t\mathbf{x}_t^\top w)})^2\mathbf{x}_t^\top\mathbf{x}_t \le \|\mathbf{x}_t\|_2^2 \le u^2$. The last inequality is by combining Lemma 1 and Lemma 3. □

A Proof of Theorem 1

Theorem 1. *Let* $\delta \in (0, 1/e)$ *and assume* $T \ge 4$*. If* $\eta_t = \frac{2}{\beta l t}$*, then with probability at least* $1 - \delta$*, we have* $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \le \frac{624u^2\log(T\log(T)/\delta)+4u^2}{\beta^2 l^2 t}$*, for any* $t \le T$*, where* $\beta \le \frac{1}{2(1+\exp(R))}$*,* l *and* u *are the lower and upper bound of the L2 norm of any* $\mathbf{x} \in \mathcal{D}$ *respectively.*

Proof. $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\Pi_R(\mathbf{w}_t - \eta_t\nabla f_t(\mathbf{w}_t)) - \mathbf{w}^*\|^2 \le \|\mathbf{w}_t - \eta_t\nabla f_t(\mathbf{w}_t) - \mathbf{w}^*\|^2 = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t\langle\nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t\langle\nabla\bar{f}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle\nabla\bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$\le \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t(\bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) + \frac{\beta l}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2) + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle\nabla\bar{f}_t(\mathbf{w}_t) -$

Submitted to 29th Conference on Neural Information Processing Systems (NIPS 2016). Do not distribute.

$\nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t(D_{KL}(p_{\mathbf{w}^*}||p_{\mathbf{w}}) + \frac{\beta l}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2) + \eta_t^2\|\nabla f_t(\mathbf{w}_t)\|^2 + 2\eta_t\langle\nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle$
$\leq (1 - \eta_t\beta l)\|\mathbf{w}_t - \mathbf{w}^*\|^2 + 2\eta_t\langle\nabla \bar{f}_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 u^2$

In the above, the first inequality is the known property of projection. The second inequality is due to Lemma 1. Taking the conditional expectation on Lemma 1 and rearranging it leads $\nabla \bar{f}_t(\mathbf{w}_t)^\top(\mathbf{w}_t - \mathbf{w}^*) \geq \bar{f}_t(\mathbf{w}_t) - \bar{f}_t(\mathbf{w}^*) + \frac{\beta}{2}((\mathbf{w}_t - \mathbf{w}^*)^\top \mathbf{x}_t)^2$. By combining the lower bound assumption on $\|\mathbf{x}\|_2$, we have the inequality. The subsequent equality is due to Lemma 3. The last inequality is because $\|\nabla f_t(\mathbf{w}_t)\|^2 \leq u^2$ and the fact that KL-divergence is positive.

Unwinding the derived inequality till $t = 2$, we get $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{2}{\beta l}\Sigma_{i=2}^t \frac{1}{i}(\Pi_{j=i+1}^t(1 - \frac{2}{t}))\langle\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^*\rangle + \frac{u^2}{\beta^2 l^2}\Sigma_{i=2}^t \frac{1}{i^2}(\Pi_{j=i+1}^t(1 - \frac{2}{t}))$. With further simplification, one can show that

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{2}{\beta l t(t-1)}\Sigma_{i=2}^t(i-1)\langle\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^*\rangle + \frac{u^2}{\beta^2 l^2 t}. \quad (1)$$

We continue to provide the bound of the distance based on (1). The following analysis now is basically the same as [1] for deriving their Proposition 1. Observe that $Z_i = (i-1)\langle\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i), \mathbf{w}_i - \mathbf{w}^*\rangle$ for each i is a martingale difference sequence, The conditional expectation given the previous rounds is 0. Furthermore, $|Z_i| \leq (t-1)\|\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i)\|_2\|\mathbf{w}_i - \mathbf{w}^*\|_2 \leq 2(t-1)u\|\mathbf{w}_i - \mathbf{w}^*\|_2 \leq 4(t-1)\frac{u^2}{\beta l}$, where the last inequality is due to Lemma 4. Thus, the expected value of Z_i is bounded. Let $\mathcal{F}_{i-1}$ be the randomness up to round $i-1$. The conditional variance $\text{Var}[Z_i|\mathcal{F}_{i-1}]$ is bounded by $(i-1)^2\|\nabla \bar{f}_i(\mathbf{w}_i) - \nabla f_i(\mathbf{w}_i)\|_2^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2 \leq 4u^2(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2$, using that fact that $\text{Var}[\cdot] \leq \mathbb{E}[(\cdot)^2]$ and Cauchy-Schwarz inequality.

Then, we can follow [1] using the lemma below, which is a variant of Freeman's inequality.

Lemma 5. *(Lemma 3 in [1]) Let $Z_1, \ldots, Z_T$ be a martingale difference sequence with a uniform bound $|Z_i| \leq b$ for all i. Let $V_s = \Sigma_{t=1}^s Var_{t-1}(Z_t)$ be the sum of conditional variance of Z_t's. Further, let $\sigma_s = \sqrt{V_s}$. Then we have, for any $\delta \leq \frac{1}{e}$ and $T \geq 4$,*
$Pr(\Sigma_{t=1}^s Z_t \geq 2\max(2\sigma_s, b\sqrt{\log(1/\delta)})\sqrt{\log(1/\delta)}$ for some $s \leq T) \leq \log(T)\delta$

By using the above analysis and Lemma 5, we have $\Sigma_{i=2}^t Z_i \leq 2\max(4u\sqrt{\Sigma_{i=2}^t(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2}, \frac{4u^2(t-1)}{\beta l}\sqrt{\log(\frac{T\log(T)}{\delta})})\sqrt{\log(\frac{T\log(T)}{\delta})}$ for all $t \leq T$ with probability $1-\delta$. Substituting it into (1) leads to $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{16}{\beta l t(t-1)}\max(u\sqrt{\Sigma_{i=2}^t(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2}, u^2\frac{(t-1)}{\beta l}\sqrt{\log(\frac{T\log(T)}{\delta})})\sqrt{\log(\frac{T\log(T)}{\delta})} + \frac{u^2}{\beta^2 l^2 t}$
$\leq \frac{16u\sqrt{\log(\frac{T\log(T)}{\delta})}}{\beta l t(t-1)}\sqrt{\Sigma_{i=2}^t(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2} + \frac{u^2}{\beta^2 l^2 t}(16\log(\frac{T\log(T)}{\delta}) + 1)$.

Let $m = \frac{16u\sqrt{\log(\frac{T\log(T)}{\delta})}}{\beta l}$, $n = \frac{u^2}{\beta^2 l^2}(16\log(\frac{T\log(T)}{\delta}) + 1)$. We can rewrite it as $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{m}{t(t-1)}\sqrt{\Sigma_{i=2}^t(i-1)^2\|\mathbf{w}_i - \mathbf{w}^*\|_2^2} + n/t$. Using mathematical induction by assuming $\|\mathbf{w}_t - \mathbf{w}^*\|^2 \leq a/t$ and finding a, we can derive the theorem. The process is the same in [1]. We need to find an a so that

$$\frac{a}{t+1} \geq \frac{m}{t(t-1)}\sqrt{\Sigma_{i=2}^t(i-1)^2\frac{a}{i}} + \frac{n}{t}.$$

It follows that $a \geq \frac{9m^2}{4} + 3n$ (c.f. proof of proposition 1 in [1]). Substituting the definition of m and n into a and observing that the base case $t = 1$ also satisties the inequality leads to the theorem.

□

B Proof of Theorem 3

Theorem 2. *Let $\delta \in (0, 1/e)$ and assume $T \geq 4$. If $\eta_t = \frac{1}{\beta l t}$, then with probability at least $1-\delta$, we have $\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \leq \frac{624u^2\log(T\log(T)/\delta) + 4u^2(1 + \frac{u}{\beta l}m(\log m + 2))}{\beta^2 l^2 t}$, for any $t \leq T$ of our distributed* *algorithm, where $\beta \leq \frac{1}{2(1+\exp(R))}$, l and u are the lower and upper bound of the L2 norm of any $\mathbf{x} \in \mathcal{D}$ respectively.*

Proof. $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\Pi_R(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_{t-m})) - \mathbf{w}^*\|^2 \leq \|\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_{t-m}) - \mathbf{w}^*\|^2 = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 \|\nabla f_t(\mathbf{w}_{t-m})\|^2 = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 \|\nabla f_t(\mathbf{w}_{t-m})\|^2$
$= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla \bar{f}_t(\mathbf{w}_t) + \nabla \bar{f}_t(\mathbf{w}_t) - \nabla \bar{f}_t(\mathbf{w}^*), \mathbf{w}_t - \mathbf{w}^*\rangle + 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 \|\nabla f_t(\mathbf{w}_{t-m})\|^2$ (using the fact that $\nabla \bar{f}_t(\mathbf{w}^*) = 0$)
$\leq (1 - 2\eta_t \beta l)\|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla \bar{f}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 \|\nabla f_t(\mathbf{w}_{t-m})\|^2$ (using Lemma 1 for $\langle \nabla \bar{f}_t(\mathbf{w}_t) - \nabla \bar{f}_t(\mathbf{w}^*), \mathbf{w}_t - \mathbf{w}^*\rangle$)

Now we bound the term $2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle$ as follows. $2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle \leq 2\eta_t \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m})\|_2 \|\mathbf{w}_t - \mathbf{w}^*\|_2 \leq 2\eta_t \frac{2u}{\beta l} \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m})\|_2 \leq \eta_t \frac{4u}{\beta l} L u \|\mathbf{w}_t - \mathbf{w}_{t-m}\|_2$, where the first inequality is due to Cauchy-Schwarz inequality, the second one is by Lemma 4, and the last one is by the Lipschitz property of the logistic function f_t (Lemma 2). For the term $\|\mathbf{w}_t - \mathbf{w}_{t-m}\|_2$, it can further be bounded by $\Sigma_{s=0}^{m-1} \|\mathbf{w}_{t-s} - \mathbf{w}_{t-s-1}\|_2 \leq \Sigma_{s=0}^{m-1} \|\Pi_R(\mathbf{w}_{t-s-1} - \eta_{t-s-1} \nabla f_t(\mathbf{w}_{t-s-1-m})) - \mathbf{w}_{t-s-1}\|_2$
$\leq u \Sigma_{s=0}^{m-1} \eta_{t-s-1} = \frac{u}{\beta l t}(\frac{t}{t-1} + \cdots + \frac{t}{t-m}) = \frac{u}{\beta l t}(m + \Sigma_{s=1}^{m} \frac{s}{t-s}) \leq \frac{u}{\beta l t}(m + m\Sigma_{s=1}^{m} \frac{1}{t-s})$
$\leq \frac{u}{\beta l t} m(\log m + 2) = \eta_t u m(\log m + 2)$. Therefore, $2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-m}), \mathbf{w}_t - \mathbf{w}^*\rangle \leq \eta_t^2 \frac{u^3}{\beta l} m(\log m + 2)$, where we have substituted L with $1/4$.

We now have $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq (1 - 2\eta_t \beta l)\|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{w}_t) - \nabla \bar{f}_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^*\rangle + \eta_t^2 u^2 (1 + \frac{u}{\beta l} m(\log m + 2))$. Following the proof of Theorem 1 leads to the results. □

C Combinatorial generalized stochastic linear bandit

In this section, we consider a case that the learner can recommend more than one item in each round. We further assume the decision set is finite, $\mathcal{D}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \cdots, \mathbf{x}_{|\mathcal{D}_t|,t}\}$, where each $\mathbf{x}_{i,t}$ is the feature representation of an item. The learner can pick k items from $\mathcal{D}_t$ in each round, where $k \geq 1$. The goal of the learner should be choosing the best subset of k items that can maximize the number of clicks. Let $\mathcal{B}_t$ be the set of items choosing by the learner in round t. The natural definition of regret would be $\Sigma_t(\max_{\mathcal{A}_t \in \mathcal{D}_t, |\mathcal{A}_t| = k} \Sigma_{\mathbf{x} \in \mathcal{A}_t} \frac{\exp(\mathbf{x}^\top \mathbf{w}^*)}{1+\exp(\mathbf{x}^\top \mathbf{w}^*)} - \Sigma_{\mathbf{x} \in \mathcal{B}_t} \frac{\exp(\mathbf{x}^\top \mathbf{w}^*)}{1+\exp(\mathbf{x}^\top \mathbf{w}^*)})$. Yet, we instead bound the following quantity

$$\Sigma_t(\max_{\mathcal{A}_t \in \mathcal{D}_t, |\mathcal{A}_t| = k} \Sigma_{\mathbf{x} \in \mathcal{A}_t} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{\mathbf{x} \in \mathcal{B}_t} \mathbf{x}^\top \mathbf{w}^*). \tag{2}$$

As before, one can show that both quantities are at the same order. The following theorem shows that Algorithm 1 achieves $\mathcal{O}(k\sqrt{T \log T})$ regret.

Algorithm 1 Combinatorial generalized stochastic linear bandit

```
Require $\gamma_t$ and $\eta_t$.
Initialization: Let $\mathbf{w_1} = \mathbf{0} \in \mathbb{R}^d$
1:  Computes a score $\mathbf{w}_t^\top \mathbf{x} + \sqrt{\gamma_t}\|\mathbf{x}\|_2$ for each item $\mathbf{x} \in \mathcal{D}_t$
2:  Select the subset of $k$ items $\mathcal{B}_t$ that has the highest scores. Observe reward $r = 1$ or $r = -1$
    for each item $\mathbf{x} \in \mathcal{B}_t$.
3:  Update $\mathbf{w}_{t+1} = \Pi(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t))$, where $\nabla f_t(\mathbf{w}_t) = \frac{1}{k}\Sigma_{\mathbf{x} \in \mathcal{B}_t} \nabla \log(1+\exp(-r\mathbf{x}^\top \mathbf{w}_t))$
```

Theorem 3. *Set $\gamma_t = \frac{624u^2 \log(T\log(T)/\delta) + 4u^2}{\beta^2 l^2 t}$ and $\eta_t = \frac{2}{\beta l t}$ in our algorithm. Let $\delta \in (0, 1/e)$ and assume $T \geq 4$. Then, with probability at least $1 - \delta$, we have (2) $\leq 2k\sqrt{\frac{624u^4 \log(T\log(T)/\delta) + 4u^4}{\beta^2 l^2}(\log(T) + 1)T}$.*

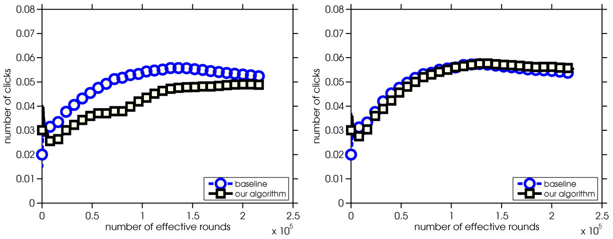
Proof. Let $\mathcal{A}_t^*$ be the subset that maximizes $\max_{\mathcal{A}_t \in \mathcal{D}_t, |\mathcal{A}_t| = k} \Sigma_{\mathbf{x} \in \mathcal{A}_t} \mathbf{x}^\top \mathbf{w}^*$. Then, $\Sigma_t(\max_{\mathcal{A}_t \in \mathcal{D}_t, |\mathcal{A}_t| = k} \Sigma_{\mathbf{x} \in \mathcal{A}_t} \mathbf{x}^\top \mathbf{w}^* - \Sigma_{\mathbf{x} \in \mathcal{B}_t} \mathbf{x}^\top \mathbf{w}^*) \leq \Sigma_t(\Sigma_{x \in \mathcal{A}_t^*} \mathbf{x}^\top \mathbf{w}_t + \sqrt{\gamma_t}\|\mathbf{x}\|_2 - \Sigma_{\mathbf{x} \in \mathcal{B}_t} \mathbf{x}^\top \mathbf{w}^*)$

$$\leq \Sigma_t(\Sigma_{\mathbf{x}\in\mathcal{B}_t}\mathbf{x}^\top\mathbf{w}_t + \sqrt{\gamma_t}\|\mathbf{x}\|_2 - \mathbf{x}^\top\mathbf{w}^*) \leq \Sigma_t 2k\sqrt{\gamma_t}\|\mathbf{x}\|_2 \leq \sqrt{\Sigma_t\gamma_t}\sqrt{\Sigma_t 4k^2\|\mathbf{x}\|_2^2} \leq 2k\sqrt{\frac{624u^4\log(T\log(T)/\delta)+4u^4}{\beta^2 l^2}}(\log(T)+1)T.$$

The first inequality is because for every $\mathbf{x}\in\mathcal{D}_t$, $\mathbf{x}^\top\mathbf{w}_t+\sqrt{\gamma_t}\|\mathbf{x}\|_2-\mathbf{x}^\top\mathbf{w}^*=(\mathbf{w}_t-\mathbf{w}^*)^\top\mathbf{x}+\sqrt{\gamma_t}\|\mathbf{x}\|_2\geq-\|\mathbf{w}_t-\mathbf{w}^*\|_2\|\mathbf{x}\|_2+\sqrt{\gamma_t}\|\mathbf{x}\|_2\geq-\sqrt{\gamma_t}\|\mathbf{x}\|_2+\sqrt{\gamma_t}\|\mathbf{x}\|_2=0$, as $\|\mathbf{w}_t-\mathbf{w}^*\|_2\leq\sqrt{\gamma_t}$ with probability $1-\delta$. The second inequality is due to line 1 in Algorithm 1 that the subset $\mathcal{B}_t$ maximizes the scores. □

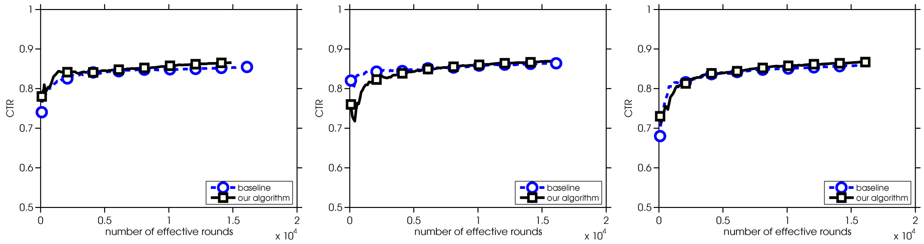
D Expereiments: addtional results

Figure 1 and Figure 2 show the CTR with respect to the number of effective rounds on Yahoo and MovieLens dataset respectively.



(a) dimensions=36 (order=1) (b) dimensions=576 (order=4)

Figure 1: Click-through rate (CTR) v.s. number of rounds on Yahoo dataset.



(a) dimensions=100 (b) dimensions=500 (c) dimensions=1000

Figure 2: Click-through rate (CTR) v.s. number of rounds on MovieLens dataset.

References

[1] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *ICML*, 2012.

[2] L. Zhang, T. Yang, R. Jin, and Z. Zhou. Online stochastic linear optimization under one-bit feedback. *ICML*, 2016.