Learning by Exploration with Information Advantage

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Huazheng Wang

August 2021

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Huazheng Wang

This Dissertation has been read and approved by the examing committee:

Advisor: Hongning Wang

Advisor:

Committee Member: David Evans

Committee Member: Haifeng Xu

Committee Member: Cong Shen

Committee Member: Mengdi Wang

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

August 2021

Abstract

Learning is a predominant theme for any intelligent system, humans, or machines. Moving beyond the classical paradigm of learning from past experiences, e.g., offline supervised learning from given labels, an intelligent learner needs to actively collect human feedback to learn from the unknowns, i.e., learning through exploration. The growing need for interactive intelligent systems in practice, such as recommender systems, smart homes, conversational systems and self-driving cars, urges the research in the *learning by exploration* paradigm. The thesis focuses on this key ingredient in interactive online learning problems, with the goal of designing algorithms that efficiently interact with and learn from human feedback in real-world environments.

There are several challenges in realizing this goal, including 1) *huge exploration space*, which is due to the large number of candidate actions and agents (users) and is typical in a practical recommender system; 2)*missing information*, where informative information regarding the actions, users and the environments may be unavailable to the intelligent system; and 3) *privacy and security concerns*, which requires a trade-off between the intelligent system's efficiency and its privacy and security guarantee. The key insight to overcome the challenges is that the *information advantage*, i.e., leveraging additional information regarding the structure of the problem such as social connectivity and context structure, offers a unique opportunity to develop advanced intelligent systems.

Based on this insight, we develop *efficient and trustful* interactive online learning systems in this thesis from three perspectives: 1) sample efficient online learning with explicit structural information; 2) efficient exploration in implicitly structured environments; and 3) privacy and security in online learning. Our study provides a deep and thorough understanding of the benefit of leveraging structural information as an advantage and extend the application of bandit learning algorithms to practical scenarios. Rigorous theoretical analysis and extensive empirical evaluation validated the approaches' applicability in various contexts and applications. By harnessing the power of information in exploration, the proposed research has been applied to high-impact real-world problems such as interactive recommendation [1,2], search result ranking [3,4], and social influence maximization [5].

Acknowledgements

Throughout my Ph.D. study, I am very fortunate to have received tremendous support. First of all, I would like to thank my adviser, Prof. Hongning Wang, for his mentorship and guidance. His vision and passion for research greatly inspired me and helped me to build my own research taste. I am grateful for his constant support from research to life and have enjoyed every insightful discussion with him. I feel extremly lucky to be his Ph.D. student.

I would like to thank my thesis committee members: Prof. David Evans, Prof. Haifeng Xu, Prof. Cong Shen, and Prof. Mengdi Wang. Prof. David Evans is an expert in privacy and security. His insightful suggestions significantly impacted my research on privacy and security aspects in bandit learning. He inspired me not only to consider the technical novelty of my research, but more importantly, the real-world impact. I have had the fortune to work with Prof. Haifeng Xu in the past two years. I have learned so much from his sharp mind and keen insights and I thank his supports in my thesis research and career development. I thank Prof. Cong Shen for providing me detailed suggestions for my thesis and my job talk, which helped me improve the quality and the presentation of my research. Prof. Mengdi Wang is an expert in bandit and reinforcement learning. I am thankful for her suggestions to my thesis research and for offering me the chance to explore interdisciplinary research between bandit learning and other fields.

I am very fortunate to have worked with many great collaborators. Special thanks to my partner, my best collaborator and my best friend Qingyun Wu. She has been a role model to me for her dedication and rigorous attitude in research. Collaborating with her during the past six years is extremely rewarding: I have become a better researcher and also a better person. I also want to thank Prof. Quanquan Gu for the collaboration on the first paper during my Ph.D. journey and suggestions for my career. I also benefited a lot from his convex optimization course. I thank all members of our HCDM group for building a great atmosphere in research, and I enjoy having discussions and group activities with everyone in the group. I would like to thank my mentors and collaborators Tie-Yan Liu, Bin Gao, Jiang Bian, Fei Tian at Microsoft Research Asia for introducing me to the field of machine learning and artificial intelligence. The great experience working with them is an important reason for me to pursue a Ph.D. degree in machine learning. I also thank Zhe Gan, Xiaodong Liu, Jingjing Liu and Jianfeng Gao at Microsoft Research and Lanbo Zhang and Yue Lu at Twitter for the mentorship and supports during my internships. I have also worked closely with Chuanhao Li, Yiling Jia, Zhiyuan Liu, Sonwoo Kim, Eric McCord-Snook, Zhige Li, Qingyao Ai, Jiaxin Mao and many others. I thank them for the collaboration and their supports.

I am grateful for the support of the Bloomberg Data Science Ph.D. fellowship. I spent two enjoyable and inspiring summers working with researchers at Bloomberg, including Qian Zhao, Shubham Chopra, Mozhgan Azimpourkivi, Abhinav Khaitan and Anju Kambadur.

Last but not least, I want to thank my parents for their unconditional love and endless supports.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Satisfying users with various personalized needs is one of the core missions of many intelligent systems. Machine learning methods are increasingly being used to help with the decision-making process involved in this mission of intelligent systems, such as recommender systems, smart homes, conversational systems and self-driving cars. Currently, most of the machine learning methods used in intelligent systems work by first collecting data and then training a fixed model for future predictions. They can provide meaningful predictions by leveraging information demonstrated in the observed data. Moving beyond the classical paradigm of learning from past experiences, an intelligent learner needs to actively collect human feedback to learn from the unknowns, i.e., *learning through exploration*. The growing needs of interactive intelligent systems urge the research in the learning by exploration paradigm.

The *learning by exploration* paradigm is the key ingredient in many interactive online learning problems, including the multi-armed bandits and reinforcement learning problems. Multi-armed bandit (MAB) algorithms [6–9] provide a principled solution for handling the explore-exploit dilemma. Intuitively, multi-armed bandit algorithms consider different decisions as arms and their main design principle is to designate a small amount of traffic to collect feedback from the environment while improving their estimation qualities on different arms in real time. With the available side information about users or items to be presented, contextual bandits have become a reference solution [10–13]. Specifically, contextual bandits assume the expected payoff is determined by a conjecture of unknown bandit parameters and given context, which is represented as a set of features extracted from both users and recommendation candidates. Such algorithms are especially advantageous when the space of recommendation is large but the payoffs are interrelated. They have been successfully applied in many important applications, e.g., content recommendation [12, 14] and display advertising [15, 16].

There are many challenges in developing efficient interactive online learning systems with human feedback. For an intelligent system that interacts with humans, huge exploration space and missing information are two challenges as the system needs to adapt to users' idiosyncratic intentions quickly. In the meanwhile, such systems are also required to ensure privacy, security and robustness, as they collect information from humans and aid decision making for humans. The key insight and motivation in this dissertation are to leverage *structural information* underlying the learning environment to advance the algorithm design, which requires a fundamental understanding of the role of information in the learning by exploration paradigm. In sophisticated yet structured real-world environments, explicit and implicit structural information are the key to designing interactive systems. To tackle the aforementioned challenged, we leverage information advantage to develop efficient and trustful contextual bandit algorithms from the following aspects: 1) sample efficient exploration with explicit structural information; 2) efficient online learning in implicitly structured environments; and 3) privacy and security concerns of bandit learning.

1.1 Contribution

1.1.1 Efficient Online Learning with Explicit Structural Information

Real-world environments are often complex yet highly structured. For example, social connections reveal potential similarity and dependency between connected users in a recommender system, and the network structure reveals assortativity information among users in social influence maximization problems. From an optimization perspective, the structure of the gradient space allows the optimizer to regularize its path to quickly achieve the optimal result. Such structural information creates unique opportunities for us to develop new online learning algorithms with reduced sample complexity, and failing to recognize them will inevitably lead to a suboptimal solution.

Efficient exploration in collaborative environments. Leveraging the information sharing structure among learning agents offers the opportunity to reduce uncertainty during exploration and expedite the convergence of online learning. Based on this insight, We propose collaborative contextual bandit learning solutions [1, 2] for online recommendation, which utilize the information about users' social connections for collaborative learning. Built on a theoretical understanding of this collaborative structure, the propose solutions improve sample efficiency of every online learning agent in this environment. Our theoretical analysis shows that in a learning system (e.g., a recommender system) with N users, after T rounds of interactions, the propose algorithm reduces regret, e.g., makes less mistakes in recommendation, up to the order of $O(\sqrt{T}\log N)$. This theoretical improvement is also validated in extensive empirical evaluations: our algorithm improves 6% click-through rate over 45 millions user visits when applied to the Yahoo front-page news recommendation.

Efficient exploration in gradient space with structural information. Ranking system is a fundamental component in information retrieval applications such as search engines. To quickly capture users' information need and avoid expensive labeling as required in offline supervised learning, online learning to rank directly learns from implicit user feedback, such as clicks. Formulated as an optimization problem named dueling bandit gradient descent, conventional solutions all followed a *uniform sampling* strategy to explore the high-dimensional gradient space. Such exploration is obviously slow and suffers from a high variance: the regret is linear to the number of ranking features, which could be *tens of thousands* in a real-world search engine. As dueling bandit gradient descent explores in the gradient space, We propose to leverage problem-specific structures in gradient estimation with user feedback to design new variance reduction techniques for online exploration [3, 4]. The key insight of [4] is that the gradient of a ranking problem belongs to the space spanned by the feature vectors of users' examined documents, which is in a much lower dimensional space than the original feature space. We propose to project the exploratory gradient onto this spanned document space to reduce variance, and proved it enjoys a significant regret reduction: the regret is now linear to the number of documents user examined under a single query, which is typically only *a couple*. This solution can be generally applied to all existing dueling bandit gradient descent based solutions. We propose to reduce the exploration space to only the *null space* of recently poorly performed gradients, to avoid repeatedly exploring less promising directions in [3], which further accelerates the convergence of online ranker estimation.

1.1.2 Efficient Online Learning in Implicitly Structured Environments

While real-world environments are highly structured, such structural information may not be explicitly available to the learners in practice. The challenge of learning by exploration in *implicitly structured environments* requires new algorithms that can infer necessary information during the online learning process.

Online learning with implicit collaboration structure via factorization. Collaboration structure can be implicit and unavailable to learning agents in many applications. However, domain knowledge, such as low-rank structure and network assortativity, in different application scenarios could be leveraged to help regularize the collaborative effect or structural information in a learning environment. This insight inspires me to design *matrix factorization based bandit algorithms* to estimate latent factors and capture the implicit collaborative effect in a problem-specific low-rank structure. The proposed solutions reduce the sample complexity in different applications, such as online recommendation [2, 17] and social influence maximization [5].

Incentivized exploration under information gap. The traditional multi-armed bandit [7] research studies the single-party setting, where the system has a full control over which arm to pull and can trade off exploitation

and exploration for long-term optimality. However, in many real-world applications, such as recommender systems and e-commerce platforms, one often faces a *two-party* game between the system and its users, and the two parties have *different* interests. We consider the problem of incentivizing exploration for myopic users in linear bandits, where the users tend to exploit arm with the highest predicted reward instead of exploring. In order to maximize the long-term reward, the system offers compensation to incentivize the users to pull the exploratory arms, with the goal of balancing the trade-off among *exploitation, exploration and compensation*. We proposed a new and practically motivated setting where the context features observed by the user are more *informative* than those used by the system, e.g., features based on users' private information are not accessible by the system. We developed a new compensation strategy under such *information gap*, and proved that the method achieves both sublinear regret and sublinear compensation. We theoretical and empirically analyze the added compensation due to the information gap, compared with the case that the system has access to the same context features as the user, i.e., without information gap. We also studied the compensation lower bound of our problem to fully characterize the price of information gap.

1.1.3 Privacy and Security in Bandit Learning

The involvement of humans in such an interactive learning process brings in both new challenges and opportunities in *privacy and security* perspectives. It is a prominent requirement for intelligent systems to be not only supportive, but also secure and protect the privacy when interacting with humans. We utilize the structural information to balance privacy and utility and understand the security and robustness of the linear bandit algorithms, aiming to develop interactive systems that are *trustworthy* to humans.

Improving privacy-utility trade-off in structured environments. Privacy concerns have been repeatedly raised as a critical issue on machine learning algorithms. Real-world privacy breaches have been reported in Amazon's and Facebook's recommender systems [18, 19], where an adversary extracts private information about a user solely based on the system's recommendation sequence. We propose a framework for private collaborative contextual bandit algorithms [20] under the notion of *global and local differential privacy*: users' feedback cannot be differentiated by an adversary from observing the interaction history. The standard approach to achieve differential privacy is by injecting noise in each individual's model to obfuscate the result, at the cost of poor recommendation quality. Our main idea is to have the system utilize the information advantage, i.e., the user-user collaboration structure, to preserve more utilities. Our solution achieves a better privacy-utility trade-off in the collaborative environment: when the users are more closely related, more utility is preserved. This research sheds light on the study of understanding the optimal balance between privacy and utility in an interactive recommender system given structural information.

Data poisoning attack on linear bandits. Are bandit algorithms vulnerable to data poisoning attacks? Recent research [21, 22] provide an affirmative answer showing that an adversary can force a non-contextual bandit algorithm to pull a target (suboptimal) arm linear times only using logarithmic costs. However, it is unclear whether linear stochastic bandits is attackable in general. Our study shows that the attackability of linear bandits is determined by the structure of the context features. Based on this insight, we proposed an efficient data poisoning attack method to manipulate the behaviour of a linear bandit algorithm when the environment is vulnerable, and showed that the difficulty in attacking a linear bandit algorithm is related to the geometry of the context. Understanding the attackability of bandit algorithms offers insights to design more robust online learning methods, and is an important step toward trustworthy interactive systems.

1.2 Dissertation Structure

The rest of this thesis is organized as follows. In Chapter 2, we describe the solutions of leveraging explicit structural information for sample efficient exploration. We propose collaborative linear bandits which utilize social influence in recommender systems and document space projection for online learning to rank in the chapter. In Chapter 3, we study learning by exploration in implicitly structured environments. We propose methods that recover latent factors in a low-rank environment and discuss how to incentivize myopic users to explore with less information. In Chapter 4, we consider privacy and security aspects of bandit learning. We equip our collaborative linear bandit algorithms with global and local differential privacy guarantees. We also show the potential vulnerability of linear bandits to data poisoning attacks. In Chapter 5, we conclude this dissertation and discuss future research directions.

Chapter 2

Sample Efficient Exploration with Explicit Structural Information

Real-world environments are often complex yet highly structured. For example, social connections reveal potential similarity and dependency between connected users in a recommender system. From an optimization perspective, the structure of the gradient space allows the optimizer to regularize its path to quickly achieve the optimal result. Such structural information creates unique opportunities for us to develop new online learning algorithms with reduced sample complexity. In this chapter, we first present our work on collaborative linear bandits for recommender system which leverage user dependency structure observed from social networks. The sample complexity is reduced according to the structure of the user connectivity. We then introduce document space projection for dueling bandit based online learning to rank, where we identify the low-rank gradient space of the ranking problem and design efficient algorithms that only explore in the reduced gradient space.

2.1 Exploration in Collaborative Environments

In this work, we develop a collaborative contextual bandit algorithm that explicitly models the underlying dependency among users. In our solution, a weighted adjacency graph is constructed, where each node represents a contextual bandit deployed for a single user and the weight on each edge indicates the influence between a pair of users. Based on this dependency structure, the observed payoffs on each user are assumed to be determined by a mixture of neighboring users in the graph. We then estimate the bandit parameters over all the users in a collaborative manner: both context and received payoffs from one user are propagated across the whole graph in the process of online updating. The proposed collaborative bandit algorithm establishes a bridge to share information among heterogeneous users and thus reduce the sample complexity of preference learning. We rigorously prove that our collaborative bandit algorithm achieves a remarkable reduction of upper regret bound with high probability, comparing to the linear regret with respect to the number of users if one simply runs independent bandits on them. Extensive experiment results on both simulations and large-scale real-world datasets verified the improvement of the proposed algorithm compared with several state-of-the-art contextual bandit algorithms. In particular, our algorithm greatly alleviates the cold-start challenge, in which encouraging performance improvement is achieved on new users and new items.

2.1.1 Related Work

The idea of modeling dependency among bandits has been explored in prior research [11, 23–26]. Studies in [27, 28] explore contextual bandits with assumptions about metric or probabilistic dependencies on the product space of context and actions. Hybrid-LinUCB [12] is such an instance, which uses a hybrid linear model to share observations across users. Social network structures are explored in bandit algorithms for introducing possible dependencies [24, 25]. In [23], parallel context-free K-armed bandits are coupled by the

social network structure among the users, where the observed payoffs from neighboring nodes are shared as side-observations to help estimate individual bandits. Besides utilizing existing social networks for modeling relatedness among bandits, there is also work automatically estimates the bandit parameters together with the dependency relation among them, such as clustering the bandits via the learned model parameters during online updating [25]. Some recent work incorporates collaboration among bandits via matrix factorization based collaborative filtering techniques: Kawale et al. preformed online matrix factorization based recommendation via Thompson sampling [29], and Zhao et al. studied interactive collaborative filtering via probabilistic matrix factorization [30]. GOB.Lin [11] requires connected users in a network to have similar bandit parameters via a graph Laplacian based model regularization. As a result, GOB.Lin explicitly requires the learned bandit parameters across related users to be close to each other.

2.1.2 Problem Formulation

In a multi-armed bandit problem, a learner takes turns to interact with the environment with a goal of maximizing its accumulated reward collected from the environment over time T. At round t, the learner makes a choice a_t among a finite, but possibly large, number of arms, i.e., $a_t \in \mathcal{A} = \{a_1, a_2, \ldots, a_K\}$, and gets the corresponding reward r_{a_t}. In the contextual bandit setting, each arm a is associated with a feature vector $\mathbf{x}_a \in \mathbb{R}^d$ ($\|\mathbf{x}_a\|_2 \leq 1$ without loss of generality) summarizing the side-information about it at a particular time point. The reward of each arm is usually assumed to be governed by a conjecture of unknown bandit parameter $\boldsymbol{\theta} \in \mathbb{R}^d$ ($\|\boldsymbol{\theta}\|_2 \leq 1$ without loss of generality), which characterizes the environment. This can be specified by a reward mapping function, say $f_{\boldsymbol{\theta}}$: $r_{a_t} = f_{\boldsymbol{\theta}}(\mathbf{x}_{a_t})$. The learner's goal of maximizing the accumulated reward can also be equivalently considered as minimizing the accumulated *regret* with respect to the oracle arm selection strategy. In particular, the accumulated T-trial regret is defined formally as,

$$\mathbf{R}(T) = \sum_{t=1}^{T} R_t = \sum_{t=1}^{T} (\mathbb{E}[r_{a_t^*}] - \mathbb{E}[r_{a_t}]) \tag{2.1}$$

where $a_t^* = \arg\max_a \mathbb{E}[r_{a,t}]$ is the best arm to display at trial t according to the oracle strategy, $r_{a_t^*}$ is the corresponding optimal reward, and $R_t := \mathbb{E}[r_{a_t^*}] - \mathbb{E}[r_{a_t}]$ is the regret at trial t.

In standard linear contextual bandit problems, the payoffs of each arm with respect to different users are assumed to be governed by a noisy version of an unknown linear function of the context vectors [10, 12]. Specifically, each user u_i is assumed to associate with an unknown parameter $\boldsymbol{\theta}_i \in \mathbb{R}^d$ (with $\|\boldsymbol{\theta}_i\| \leq 1$), which determines the payoff of a_t by $r_{a_t,i} = \mathbf{x}_{a_t,i}^{\mathsf{T}} \boldsymbol{\theta}_i + \epsilon_t$, where the random variable ϵ_t is drawn from a Guassian distribution $N(0, \sigma^2)$. $\boldsymbol{\theta}$s are independently estimated based on the observations from each individual user. However, due to the existence of mutual influence among users, an isolated bandit can hardly explain all the observed payoffs even for a single user. For example, the context vectors fail to encode such dependency. To capitalize on the additional information embedded in the dependency structure among users (i.e., $\boldsymbol{\theta}$ for different users), we propose to study contextual bandit problems in a collaborative setting.

In this collaborative environment, we place the bandit algorithms on a weighted graph $G = (V, E)$, which encodes the affinity relationship among users. Specifically, each node $v_i \in \{V_1, ..., V_N\}$ in G hosts a bandit parameterized by $\boldsymbol{\theta}_i$ for user i; and the edges in E represent the affinity relation over pairs of users. This graph can be described as an $N \times N$ stochastic matrix $\mathbf{W}$. In this matrix, each element w_{ij} is nonnegative and proportional to the influence that user i has on user j in determining the payoffs of different arms. $w_{ij} = 0$ if and only if user i has no influence on user j. $\mathbf{W}$ is normalized such that $\sum_{i=1}^{N} w_{ij} = 1$ for $j \in \{1, ..., N\}$ (the sum of each column is 1). In this work, we assume $\mathbf{W}$ is time-invariant and known to the learner beforehand.

Based on the graph G, collaboration among bandits happens when determining the payoff of a particular arm with respect to a given user. To denote this, we define a $d \times N$ matrix $\boldsymbol{\Theta}$, which consists of parameters from all the bandits in the graph: $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$. Accordingly, we define a context feature matrix $\mathbf{X}_t = (\mathbf{x}_{a_{t,1}}, \ldots, \mathbf{x}_{a_{t,N}})$, where the ith column is the context vector $\mathbf{x}_{a_{t,i}}$ for arm a at trial t selected for user i. The collaboration among bandits characterized by the influence matrix $\mathbf{W}$ results in a new bandit parameter

Algorithm 1 Collaborative li2010contextual

Inputs: $\alpha \in \mathbb{R}_+$,$\lambda \in [0,1]$, $\mathbf{W} \in \mathbb{R}^{N \times N}$
Initialize: $\mathbf{A}_1 \leftarrow \lambda \mathbf{I}$, $\mathbf{b}_1 \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\vartheta}}_1 \leftarrow \mathbf{A}_1^{-1}\mathbf{b}_1, \mathbf{C}_1 \leftarrow (\mathbf{W}^\mathsf{T} \otimes \mathbf{I})\mathbf{A}_1^{-1}(\mathbf{W} \otimes \mathbf{I})$,
for $t = 1$ to T **do**
    Receive user u_t
    Observe context vectors, $\mathbf{x}_{a_t,u_t} \in \mathbb{R}^d$ for $\forall a \in \mathcal{A}$
    Take action $a_t = \arg\max_{a \in \mathcal{A}} \mathring{\mathcal{X}}_{a_t,u_t}^\mathsf{T} \text{Vec}(\widehat{\boldsymbol{\Theta}}_t \mathbf{W}) + \alpha \sqrt{\mathring{\mathcal{X}}_{a_t,u_t}^\mathsf{T} \mathbf{C}_t \mathring{\mathcal{X}}_{a_t,u_t}}$
    Observe payoff r_{a_t,u_t}
    $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T})\text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}$
    $\mathbf{b}_{t+1} \leftarrow \mathbf{b}_t + \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T}) r_{a_t,u_t}$
    $\mathbf{C}_{t+1} \leftarrow (\mathbf{W}^\mathsf{T} \otimes \mathbf{I})\mathbf{A}_{t+1}^{-1}(\mathbf{W} \otimes \mathbf{I})$
    $\hat{\boldsymbol{\vartheta}}_{t+1} \leftarrow \mathbf{A}_{t+1}^{-1}\mathbf{b}_{t+1}$

matrix $\bar{\Theta} = \boldsymbol{\Theta}\mathbf{W}$, which determines the payoff r_{a_t,u_t} of arm a_t for user u_t at trial t by,

$$r_{a_t,u_t} - diag_t(\mathbf{X}_t^\mathsf{T}\boldsymbol{\Theta}\mathbf{W}) \sim N(0, \sigma^2) \tag{2.2}$$

where $diag_t(\mathbf{X})$ is the operation returning the t-th element in the diagonal of matrix $\mathbf{X}$.

Eq (2.2) postulates our *additive* assumption about reward generation in this collaborative environment: the reward r_{a_t,u_t} is not only determined by user u_t's own preference on the arm a_t (i.e., $w_{u_t u_t}\mathbf{x}_{a_t,u_t}^\mathsf{T}\boldsymbol{\theta}_{u_t}$), but also by the judgements from the neighbors who have influence on u_t (i.e., $\sum_{j \neq u_t} w_{u_t j}\mathbf{x}_{a_t,j}^\mathsf{T}\boldsymbol{\theta}_j$). This enables us to distinguish a user's intrinsic preference of the recommended content from his/her neighbors' influence, i.e., personalization. In addition, the linear payoff assumption in our model is to simplify the discussion in this work; and it can be relaxed via a generalized linear model [13] to deal with nonlinear rewards.

We should note that our model assumption about the collaborative bandits is different from that specified in the GOB.Lin model [11]. In GOB.Lin, connected users in the graph are required to have similar underlying bandit parameters, i.e., via graph Laplacian regularization over the learned bandit parameters. And their assumption about reward generation follows conventional contextual bandit settings, i.e., rewards are independent across users. In our setting, neighboring users do not have to share similar bandit parameters, but they will generate influence on their neighbors' decisions. This assumption is arguably more general, and it leads to an improved upper regret bound and practical performance. Theoretical comparison between these two algorithms will be rigorously discussed in Section 3.1.3.

2.1.3 Collaborative Linear Bandit Algorithm

To simplify the notations in our following discussions, we define two long context feature vectors and a long bandit parameter vector based on the vectorize operation Vec$(\cdot)$. We define $\mathcal{X}_{a_t} = \text{Vec}(\mathbf{X}_{a_t}) = (\mathbf{x}_{a_t,1}^\mathsf{T}, \ldots, \mathbf{x}_{a_t,N}^\mathsf{T})^\mathsf{T}$, which is a concatenation of context feature vectors of the chosen arm a_t at trial t for all the users. And we define $\mathring{\mathcal{X}}_{a_t,u_t} = \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t})$, in which $\mathring{\mathbf{X}}_{a_t,u_t}$ is a special case of $\mathbf{X}_{a_t}$: only the column corresponding to the user u_t at time t is set to $\mathbf{x}_{a_t,u_t}^\mathsf{T}$, and all the other columns are set to zero. This corresponds to the situation that at trial t the learner only needs to interact with one user. Correspondingly, we define $\boldsymbol{\vartheta} = \text{Vec}(\boldsymbol{\Theta}) = (\boldsymbol{\theta}_1^\mathsf{T}, \boldsymbol{\theta}_2^\mathsf{T}, ..., \boldsymbol{\theta}_N^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{dN}$ as the concatenation of bandit parameter vectors over all the users.

With the collaborative assumption about the expected payoffs defined in Eq (2.2), we appeal to ridge regression for estimating the unknown bandit parameter $\boldsymbol{\theta}$ for each user. In particular, we simultaneously estimate the global bandit parameter matrix $\boldsymbol{\Theta}$ for all the users as follows,

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{2}\sum_{t=1}^{T}(\mathring{\mathcal{X}}_{a_t,u_t}^\mathsf{T}\text{Vec}(\boldsymbol{\Theta}_t\mathbf{W}) - r_{a_t,u_t})^2 + \frac{\lambda}{2}tr(\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta}) \tag{2.3}$$

where $\lambda \in [0, 1]$ is a trade-off parameter of L2 regularization in ridge regression.

Since the objective function defined in Eq (2.3) is quadratic with respect to $\mathbf{\Theta}$, we have a closed-form estimation of $\mathbf{\Theta}$ as $\hat{\boldsymbol{\vartheta}}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$, in which $\hat{\boldsymbol{\vartheta}} = \text{Vec}(\widehat{\mathbf{\Theta}})$ and $\mathbf{A}_t$ and $\mathbf{b}_t$ are computed as,

$$\mathbf{A}_t = \lambda\mathbf{I} + \sum_{t'=1}^{t} \text{Vec}(\mathring{\mathbf{X}}_{a_{t'},u_{t'}}\mathbf{W}^\mathsf{T})\text{Vec}(\mathring{\mathbf{X}}_{a_{t'},u_{t'}}\mathbf{W}^\mathsf{T})^\mathsf{T} \tag{2.4}$$

$$\mathbf{b}_t = \sum_{t'=1}^{t} \text{Vec}(\mathring{\mathbf{X}}_{a_{t'},u_{t'}}\mathbf{W}^\mathsf{T})r_{a_{t'},u_{t'}} \tag{2.5}$$

where $\mathbf{I}$ is a $dN \times dN$ identity matrix.

The effect of collaboration among bandits is clearly depicted in the above estimation of $\mathbf{\Theta}$. Matrix $\mathbf{A}_t$ and vector $\mathbf{b}_t$ store global information shared among all the bandits in the graph. More specifically, the context vector $\mathbf{x}_{a_t,u_t}$ and payoff r_{a_t,u_t} observed in user u_t at trial t are propagated through the whole graph via the relational matrix $\mathbf{W}$. To understand this, note that $\text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T})$ is a dense vector with projected context vectors on every user, while the original $\mathring{\mathcal{X}}_{a_t,u_t}$ is a sparse vector with observations only at active users u_t. Because of this information sharing, at certain trial t, although some users might generate any observation yet (i.e., cold-start), they can already start from a non-random initialization of their bandit parameters $\boldsymbol{\theta}_i$. It is easy to verify that when $\mathbf{W}$ is an identity matrix, i.e., users have no influence among each other, the estimation of $\mathbf{\Theta}$ degenerates to independently computing N different $\boldsymbol{\theta}$s (since $\text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T}) = \mathring{\mathcal{X}}_{a_t,u_t}$). And the mutual influence will be maximized when $\mathbf{W}$ is a uniform matrix, i.e., all the users have equivalent influence to each other. We have to emphasize that the benefit of this collaborative estimation of $\mathbf{\Theta}$ is not to just simply compute the $\boldsymbol{\theta}$s in an integrated manner; but because of the collaboration among users, the estimation uncertainty of all $\boldsymbol{\theta}$s can be quickly reduced comparing to simply running N independent bandit algorithms. This in turn leads to an improved regret bound. We will elaborate the effect of collaboration in online bandit learning with more theoretical justifications in Section 3.1.3.

The estimated bandit parameters $\widehat{\mathbf{\Theta}}$ predict the expected payoff of a particular arm for each user according to the observed context feature matrix $\mathbf{X}_t$. To complete an adaptive bandit algorithm, we need to design the exploration strategy for each user. Our collaborative assumption in Eq (2.2) implies that r_{a_t,u_t} across users are independent given $\mathbf{X}_t$ and $\mathbf{W}$. As a result, for any σ, i.e., the standard deviation of Gaussian noise in Eq (2.2), the following inequality holds with probability at least $1 - \delta$,

$$|r_{a_t^*,u_t} - r_{a_t,u_t}| \leq \alpha_t \sqrt{\text{Vec}(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\mathbf{A}_t^{-1}\text{Vec}(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})} \tag{2.6}$$

where α_t is a parameter in our algorithm defined in Lemma 1 of Section 3.1.3 and δ is embedded in the computation of α_t. The proof of this inequality can be found in the Appendix.

The inequality in Eq (2.6) gives us a reasonably tight upper confidence bound (UCB) for the expected payoff of a particular arm over all users in the graph G, from which a UCB-style action-selection strategy can be derived. In particular, at trial t, we choose an arm for user u_t by,

$$a_{t,u_t} = \arg\max_{a_t \in \mathcal{A}} \Big(\mathring{\mathcal{X}}_{a_t,u_t}^\mathsf{T} \text{Vec}(\widehat{\mathbf{\Theta}}_t\mathbf{W}) + \alpha_t \sqrt{\text{Vec}(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\mathbf{A}_t^{-1}\text{Vec}(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})} \Big) \tag{2.7}$$

We name this resulting algorithm as Collaborative Linear Bandit, or CoLin in short. The detailed description of CoLin is illustrated in Algorithm 1, where we use the property that $\text{Vec}(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T}) = (\mathbf{W} \otimes \mathbf{I})\text{Vec}(\mathring{\mathbf{X}}_{u_t})$ $= (\mathbf{W} \otimes \mathbf{I})\mathring{\mathcal{X}}_t$ to simplify Eq (2.7).

Another evidence of the benefit from collaboration among users is demonstrated in Algorithm 1. When estimating the confidence interval of the expected payoff for action a_t in user u_t at trial t, CoLin not only considers the prediction confidence from bandit u_t, but also that from its neighboring bandits (as described by the Kronecker product between $\mathbf{W}$ and $\mathbf{I}$). When $\mathbf{W}$ is an identity matrix, such effect disappears. Clearly, this

collaborative confidence interval estimation will help the algorithm quickly reduce estimation uncertainty, and thus leads to the optimal solution more rapidly.

One potential issue with CoLin is its computational complexity: matrix inverse has to be performed on $\mathbf{A}_t$ at every trial. First, because of the rank one update of matrix $\mathbf{A}_t$ (8th step in Algorithm 1), quadratic computation complexity is possible via applying the Sherman-Morrison formula. Second, we may compute $\mathbf{A}_t^{-1}$ in a mini-batch manner to further reduce computation with some extra penalty in regret. We will leave this as our future research.

2.1.4 Regret Analysis

In this section, we provide detailed regret analysis of our proposed CoLin algorithm. We first prove that the estimation error of bandit parameters $\widehat{\boldsymbol{\Theta}}$ is upper bounded in Lemma 1.

Lemma 1. *For any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters in CoLin is bounded by,*

$$\|\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\ln\left(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\right) - 2\ln(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\|$$

in which $\|\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} = \sqrt{(\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*)^\intercal \mathbf{A}_t (\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*)}$, i.e., the matrix norm induced by the positive semidefinite matrix $\mathbf{A}_t$.

Based on Lemma 1, we have the following theorem about the regret upper bound of the CoLin algorithm.

Theorem 1. *With probability at least $1-\delta$, the accumulated regret of CoLin algorithm satisfies,*

$$\boldsymbol{R}(T) \leq 2\alpha_T \sqrt{2dNT\ln\left(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\right)} \tag{2.8}$$

in which α_T is the upper bound of $\|\hat{\boldsymbol{\vartheta}}_T - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_T}$ and it can be explicitly calculated based on Lemma 1.

The detailed proof of this theorem is provided in the Appendix.

As shown in Theorem 1, the graph structure plays an important role in the upper regret bound of our CoLin algorithm. Consider two extreme cases. First, when $\mathbf{W}$ is an identity matrix, i.e., no influence among users, the upper regret bound degenerates to $O(N\sqrt{T}\ln\frac{T}{N})$. Second, when the graph is fully connected and uniform, i.e., $\forall i, j, w_{ij} = \frac{1}{N}$, such that users have homogeneous influence among each other, and the upper regret bound of CoLin decreases to $O(N\sqrt{T}\ln\frac{T}{N^2})$. That means via collaboration, CoLin achieves an $O(\sqrt{T}\ln N)$ regret reduction for every single user in the graph comparing to the independent case.

Note that the our regret analysis in Theorem 1 is in a very general form, in which we did not make any assumption about the order or frequency that each user will be served. To illustrate the relationship between the proposed collaborative bandit algorithm and conventional independent bandit algorithms in a more intuitive way, we can make a very specific assumption about how a sequential learner interacts with a set of users. Assuming all the users are evenly served by CoLin, i.e., each user interacts with the learner $\bar{T} = \frac{T}{N}$ times. When $\mathbf{W}$ is an identity matrix, the regret bound of CoLin degenerates to the case of running N independent li2010contextual, whose upper regret bound is $O(N\sqrt{\bar{T}}\ln\bar{T})$. When $\mathbf{W}$ is uniform, the regret bound reduces to $O(N\sqrt{\bar{T}}\ln\frac{\bar{T}}{N})$, where we achieves an $O(\sqrt{\bar{T}}\ln N)$ regret reduction comparing to running N independent li2010contextuals on each single user. The proof of regret bound in this special case is given in the Appendix.

It is necessary to compare the derived upper regret bound of CoLin with that in the GOB.Lin algorithm [11], which also exploits the relatedness among a set of users. In GOB.Lin, the divergence among every pair of

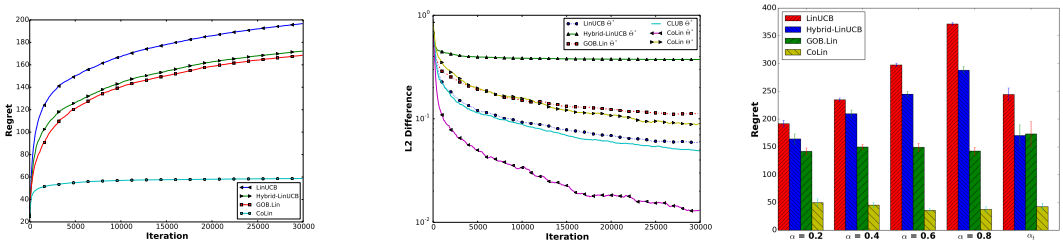
bandits (if connected in the graph) is measured by Euclidean distance between the learned bandit parameters. In its upper regret bound, such divergence is accumulated throughout the iterations. In extreme case where users are all connected but associate with totally distinct bandit parameters, GOB.Lin's upper regret bound could be much worse than running N independent bandits, due to this additive pairwise divergence. While in our algorithm, such divergence is controlled by the multiplicative factor $\sum_{t=1}^{T}\sum_{j} w_{u_t j}^2 \leq T$. We can rigorously prove the following inequalities between the upper regret bound of CoLin ($R_C(T)$) and GOB.Lin ($R_G(T)$) always holds,

$$0 \leq R_G^2(T) - R_C^2(T) \leq 16TN\ln(1+\frac{2T}{dN^2})\sum_{(i,j)\in E}\|\boldsymbol{\theta}_i^* - \boldsymbol{\theta}_j^*\|^2$$

It is clear to notice that if there is no influence between the users in the collection, i.e., no edge in G, these two algorithms' regret bound touches (both degenerate to N independent contextual bandits). Otherwise, GOB.Lin will always lead to a worse and faster growing regret bound than our CoLin algorithm.

In addition, limited by the use of graph Laplacian, GOB.Lin can only capture the binary connectivity relation among users. CoLin differentiates the strength of connections with a stochastic relational graph. This makes CoLin more general when modeling the relatedness among bandits and provides a tighter upper regret bound. This effect is also empirically verified by our experiments on both synthetic and real-world datasets.

2.1.5 Experiments



(a) Convergence of accumulated regret (b) Parameter estimation error (c) The effect of α

Figure 2.1: Analysis of regret, bandit parameter estimation and parameter tuning.

We performed empirical evaluations of our CoLin algorithm against several state-of-the-art contextual bandit algorithms, including N independent li2010contextual [12], hybrid li2010contextual with user features [12], GOB.Lin [11], and online cluster of Bandits (CLUB) [25]. Among these algorithms, hybrid li2010contextual exploits user dependency via a set of hybrid linear models over user features, GOB.Lin encodes the user dependency via graph-based regularization over the learned bandit parameters, and CLUB clusters users during online learning to enable model sharing. In addition, we also compared with several popularly used context-free bandit algorithms, including EXP3 [9], auer2002finite [8] and ϵ-greedy [8]. But their performance is much worse than the contextual bandits, and thus we do not include their performance in the following discussions.

We tested all the algorithms on a synthetic dataset via simulations, a large collection of click stream from Yahoo! Today Module dataset [12], and two real-world dataset extracted from the social bookmarking web service Delicious and music streaming service LastFM [11]. Extensive experiment comparisons confirmed the advantage of our proposed CoLin algorithm against all the baselines. More importantly, comparing to the baselines that also exploit user dependencies, CoLin performs significantly better in identifying users' preference on less popular items (items that are only observed among a small group of users). This validates that with the proposed collaborative learning among users, CoLin better alleviates the cold-start challenge comparing to the baselines.

Experiments on synthetic dataset

In this experiment, we compare the bandit algorithms based on simulations and use the accumulated regret and bandit parameter estimation accuracy as the performance metrics.

Simulation Setting. In simulation, we generate N users, each of which is associated with a d-dimensional parameter vector $\boldsymbol{\theta}^*$, i.e., $\boldsymbol{\Theta}^* = (\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_N^*)$. Each dimension of $\boldsymbol{\theta}_i^*$ is drawn from a uniform distribution $U(0,1)$ and normalized to $\|\boldsymbol{\theta}_i^*\| = 1$. $\boldsymbol{\Theta}^*$ is treated as the ground-truth bandit parameters for reward generation, and they are unknown to bandit algorithms. We then construct the golden relational stochastic matrix $\mathbf{W}$ for the graph of users by defining $w_{ij} \propto \langle \boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^* \rangle$, and normalize each column of $\mathbf{W}$ by its L1 norm. The resulting $\mathbf{W}$ is disclosed to the bandit algorithms. In the end, we generate a size-K action pool $\mathcal{A}$. Each action a in $\mathcal{A}$ is associated with a d-dimensional feature vector $\mathbf{x}_a$, each dimension of which is drawn from $U(0,1)$. We also normalize $\mathbf{x}_a$ by its L1 norm. To construct user features for hybrid li2010contextual algorithm, we perform Principle Component Analysis (PCA) on the relational matrix $\mathbf{W}$, and use the first 5 principle components to construct the user features.

To simulate the collaborative reward generation process among users, we first compute $\bar{\Theta}^* = \boldsymbol{\Theta}^* \mathbf{W}$ and then compute the payoff of action a for user i at trial t as $r_{a_{t,i}} = diag_i(\mathbf{X}_t^\intercal \bar{\Theta}^*) + \epsilon_t$, where $\epsilon_t \sim N(0, \sigma^2)$. To increase the learning complexity, at each trial t, our simulator only discloses a subset of actions in $\mathcal{A}$ to the learning algorithms for selection, e.g., randomly select 10 actions from $\mathcal{A}$ without replacement. In simulation, based on the known bandit parameters $\bar{\Theta}^*$, the optimal action $a_{t,i}^*$ and the corresponding payoff $r_{a_{t,i}^*}$ for each bandit i at trial t can be explicitly computed.

Under this simulation setting, we compared hybrid li2010contextual, N independent li2010contextual, GOB.Lin, CLUB and our CoLin algorithm. In particular, at each trial t, the same set of actions are presented to all the algorithms; and the Gaussian noise ϵ_t is fixed for all those actions at trial t. In our experiments, we fixed the feature dimension d to 5, article pool size K to 1000, and set the trade-off parameter λ for L2 regularization to 0.2 in all the algorithms. We compared the regret of different bandit algorithms during adaptive learning. Furthermore, since we have the ground-truth bandit parameters available in the simulator, we also compared the quality of learned parameters in each contextual bandit algorithms. This unveils the nature of each bandit algorithm, e.g., how accurately they can recover a user's true preference.

Results & Analysis. We first set the user size N to 100 and fix the standard deviation σ to 0.1 in the Gaussian noise for reward generation. All the contextual bandit algorithms are executed up to 300 iterations per user in this experiment. We report the accumulated regret as defined in Eq (2.1) and the Euclidean distance between the learnt bandit parameters from different algorithms and the ground-truth in Figure 2.1. To reduce randomness in simulation-based evaluation, we reported the mean and standard deviation of final regret from different algorithms after 30,000 iterations over 5 independent runs for results in all following experiments. To increase visibility, we did not plot error bars in Figure 2.1 (a) and (b).

As we can find in Figure 2.1 (a), simply running N independent li2010contextual algorithm gives us the worst regret, which is expected. Hybrid li2010contextual, which exploits user dependency via a set of hybrid linear models over user features performed better than li2010contextual, but still much worse than CoLin. Although GOB.Lin also exploits the graph structure when estimating the bandit parameters, its assumption about the dependency among bandits is too restrictive to well capture the information embedded in the interaction with users. We should note that in our simulation, by multiplying the relational matrix $\mathbf{W}$ with the ground-truth bandit parameter matrix $\boldsymbol{\Theta}^*$, the resulting bandit parameters $\bar{\Theta}^*$ align with GOB.Lin's assumption, i.e., neighboring bandits are similar. And $\bar{\Theta}^*$ is used in reward generation. Therefore, our simulation does not produce any bias against GOB.Lin. In Figure 2.1 (a) we did not include CLUB, whose regret grew linearly. After looking into the selected arms from CLUB, we found because of the aggregated decisions from users in the automatically constructed user clusters, CLUB always chose suboptimal arms, which led to a linearly increasing regret.

In Figure 2.1 (b), we compared accuracy of the learnt bandit parameters from different algorithms. Because of their distinct modeling assumptions, li2010contextual, hybrid li2010contextual, CLUB and GOB.Lin cannot directly estimate $\boldsymbol{\Theta}^*$, i.e., the true bandit parameters for each user. Instead, they can only estimate $\bar{\Theta}^*$, which directly governs the generation of observed payoffs. Only CoLin can estimate both $\bar{\Theta}^*$ and $\boldsymbol{\Theta}^*$. As we can find in the results, CoLin gave the most accurate estimation of $\bar{\Theta}^*$, which partially explains its superior performance in regret. We also find that li2010contextual actually achieved a more accurate estimation of $\bar{\Theta}^*$ than GOB.Lin, but its regret is much worse. To understand this, we looked into the actual execution of li2010contextual and GOB.Lin, and found that because of the graph Laplacian regularization in GOB.Lin, it better controlled exploration in arm selection and therefore picked the optimal arm more often than li2010contextual. Hyrbid

Table 2.1: Accumulated regret with different bandit size (σ=0.1).

Bandit Size (N)	40	80	100	200
li2010contextual	75.31±5.11	168.42±9.90	191.53±6.18	355.56±7.23
Hybridli2010contextual	59.12±2.11	150.09±5.29	164.11±9.19	311.43±11.59
GOB.Lin	58.49±5.04	143.42±5.28	141.96±6.36	275.32±10.51
CoLin	**21.78**± 12.84	47.73±4.31	49.83±6.55	77.38±20.59

Table 2.2: Accumulated regret with different noise level (N=100).

Noise (σ)	0.01	0.05	0.1	0.3
li2010contextual	92.72±2.56	116.53±3.07	191.53 ±6.18	830.47±69.48
Hybridli2010contextual	69.47±2.12	91.48±1.94	164.11±9.19	759.93±39.15
GOB.Lin	58.85±6.25	82.33±1.53	141.96±6.36	708.13±43.73
CoLin	41.69±6.95	40.95±4.43	49.83±6.55	83.98±8.57

Table 2.3: Accumulated regret with different noise level on matrix $\mathbf{W}$ (N=100).

matrix noise (δ)	0	0.01	0.03	0.05
Hybridli2010contextual	164.11±9.19	171.74±7.67	171.51±13.31	163.93±9.19
GOB.Lin	141.96±6.36	163.28±5.63	164.36±6.66	169.87±11.89
CoLin	49.83±6.55	54.42±2.45	101.39±6.01	239.88±13.86

Table 2.4: Accumulated regret with different matrix sparsity level.

Sparsity (M/N)	20/100	40/100	60/100	80/100
Hybridli2010contextual	135.98±5.11	141.15±4.82	150.49±4.58	160.12±7.55
GOB.Lin	133.30±3.98	126.13±5.59	143.29±6.49	143.42±5.82
CoLin	39.74±8.80	30.76±3.66	37.29±3.55	49.56±8.88

li2010contextual's estimation of $\bar{\Theta}^*$ is the worst, but it is expected: hybird li2010contextual uses a shared model and a personalized model to fit the observations. Comparing to CoLin's estimation quality of $\bar{\Theta}^*$, its estimation of $\mathbf{\Theta}^*$ is much worse. The main reason is that CoLin has to decipher $\mathbf{\Theta}$ from the estimated $\bar{\Theta}$ based on $\mathbf{W}$, where noise is accumulated to prevent accurate estimation. Nevertheless, this result demonstrates the possibility of discovering each individual user's true preference from their compound feedback. This is meaningful for many practical applications, such as user modeling and social influence analysis. We also notice that although CLUB's estimated $\bar{\Theta}^*$ is almost as good as li2010contextual's (as shown in Figure 2.1 (b)), its regret is the worst. As we described earlier, CLUB's aggregated decision at user cluster level constantly forced the algorithm to choose sub-optimal arms; but the reward generation for each arm in our simulator follows that defined in Eq (2.2), which still provides validate information for CLUB to estimate $\bar{\Theta}^*$ with reasonable accuracy.

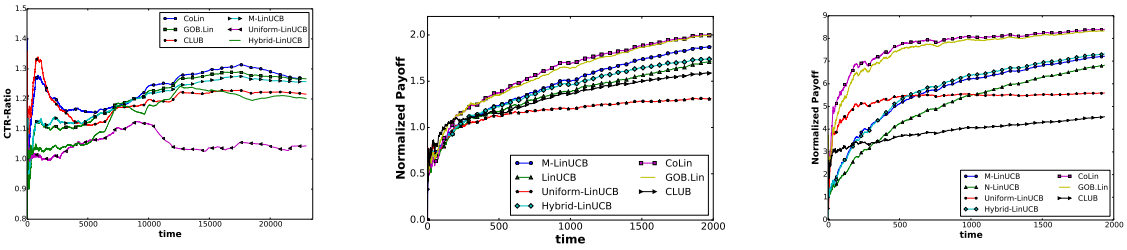
In Figure 2.1 (c), we investigated the effect of exploration parameter α_t's setting in different algorithms. The last column indexed by α_t represented the theoretical values of α computed from the algorithms' corresponding regret analysis. As shown in the results, the empirically tuned α yields comparable performance to the theoretical values, and makes online computation more efficient. As a result, in all our following experiments we will use a fixed α instead of a computed α_t.

To further investigate the convergence property of different bandit algorithms, we examined the following four scenarios: 1) various user sizes N, 2) different noise level σ, 3) a corrupted affinity matrix $\mathbf{W}$, and 4) a sparse affinity matrix $\mathbf{W}$, in reward generation. We report the results in Table 2.1 to 2.4. Because of its poor performance, we did not include CLUB in those tables. Firstly, in Table 2.1, we fixed the noise level σ to 0.1 and varied the user size N from 40 to 200. We should note in this experiment the total number of iterations varies as every user will interact with the bandit learner 300 times. The regret in li2010contextual goes linearly with respect to the number of users, since no information is shared across them. Via model sharing, hybrid li2010contextual achieved some regret reduction compared with li2010contextual; but its regret still increases linearly with the number of users. Compared with the independent bandits, we can clearly observe the regret reduction in CoLin with increasing number of users. As we discussed in Section 3.1.3, although GOB.Lin exploits the dependency among users, its regret might be even worse than running N

independent li2010contextuals, especially when the divergence between users is large. Secondly, in Table 2.2, we fixed N to 100 and varied the noise level σ from 0.01 to 0.3. We can notice that CoLin is more robust to noise in the feedback: its regret grows much slower than all baselines. Our current regret analysis does not consider the effect of noise in reward, as long as it has a zero mean and finite variance. It would be interesting to incorporate this factor in regret analysis to provide more insight of collaborative bandit algorithms.

Thirdly, in CoLin, we have assumed the adjacency matrix $\mathbf{W}$ is known to the algorithm beforehand. However, in reality one might not precisely recover this matrix from noisy observations, e.g., via social network analysis. It is thus important to investigate the robustness of collaborative bandit algorithms to a noisy $\mathbf{W}$. We fixed the user size N to 100 and corrupted the ground-truth adjacency matrix $\mathbf{W}$: add Gaussian noise $N(0, \delta)$ to w_{ij} and normalize the resulting matrix. We refer to this noisy adjacency matrix as $\mathbf{W}_0$. The simulator still uses the true adjacency matrix $\mathbf{W}$ to compute the reward of each action for a given user; while the noisy matrix $\mathbf{W}_0$ will be provided to the bandit algorithms, i.e., CoLin and GOB.Lin. This newly introduced Gaussian noise is different from the noise in generating the rewards as described in Eq (2.2).

From the accumulated regret shown in Table 2.3, we can find that under moderate noise level, CoLin performed much better than GOB.Lin; but CoLin is more sensitive to noise in $\mathbf{W}$ than GOB.Lin. Because CoLin utilizes a weighted adjacency graph to capture the dependency among users, it becomes more sensitive to the estimation error in $\mathbf{W}$. While in GOB.Lin, because only the graph connectivity is used and the random noise is very unlikely to change the graph connectivity, its performance is more stable. Further theoretical analysis of how an inaccurate estimation of $\mathbf{W}$ would affect the resulting regret will be an interesting future work yet to explore.



(a) Yahoo dataset (b) Delicious dataset (c) LastFM dataset

Figure 2.2: Normalized reward on three real-world datasets

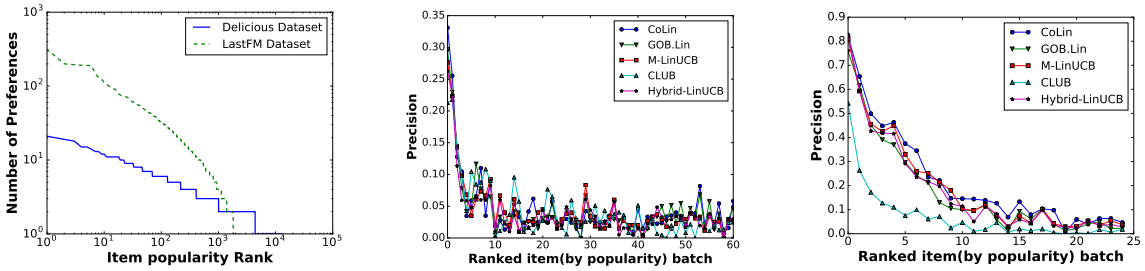
Finally, the regret analysis of CoLin shows that its upper regret bound is related to the graph structure through the term $\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2$ and GOB.Lin's regret bound is related to the graph connectivity [11]. We designed another set of simulation experiments to verify the effect of graph structure on CoLin and GOB.Lin. In this experiment, we set the user size N to 100 and controlled the graph sparsity as follows: for each user in graph G, we only included the edges from his/her top M most influential neighbors (measured by the edge weight in $\mathbf{W}$) in the adjacency matrix, and normalized the resulting adjacency matrix to a stochastic matrix. No noise is added to $\mathbf{W}$ in this experiment (i.e., $\delta = 0$).

As shown in Table 2.4, the regret of all bandit algorithms increases as $\mathbf{W}$ becomes sparser, i.e., less information can be shared across users. We can observe that the regret of CoLin increases faster than that in GOB.Lin, since more information becomes unavailable to CoLin. The results empirically verified that CoLin's regret bound is directly related to the graph structure described by the term $\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2$ and GOB.Lin's regret bound is only related to the graph connectivity.

Experiments on Yahoo! Today Module

In this experiment, we compared our CoLin algorithm with li2010contextual, hybrid li2010contextual, GOB.Lin and CLUB on a large collection of ten days' real traffic data from Yahoo! Today Module [12] using the unbiased offline evaluation protocol proposed in [31].

The dataset contains 45,811,883 user visits to the Today Module in a ten-day period in May 2009. For each logged event, both the user and each of the 10 candidate articles are associated with a feature vector of six dimensions (including a constant bias feature), which is constructed by a conjoint analysis with a



(a) Item popularity (b) Delicious dataset (c) LastFM dataset

Figure 2.3: Item-based analysis on Delicious and LastFM datasets

bilinear model [12]. However, this dataset does not contain any user identity. This forbids us to associate the observations with individual users. To address this limitation, we first clustered all users into user groups by applying K-means algorithm on the given user features. Each observation is assigned to its closest user group. The weight in the adjacency matrix $\mathbf{W}$ is estimated by the dot product between the centroids from K-means' output, i.e., $w_{ij} \propto \langle u_i, u_j \rangle$. The CoLin and GOB.Lin algorithms are then executed over those identified user groups. For the li2010contextual baseline, we tested two variants: one is individual li2010contextuals running over the identified user groups and it is denoted as M-li2010contextual; another one is a uniform li2010contextual shared by all the users, i.e., it does not distinguish individual users, and thus it is denoted as Uniform-li2010contextual.

In this experiment, click-through-rate (CTR) was used to evaluate the performance of all bandit algorithms. An algorithm's CTR is defined as the number of clicks its recommendations receive divided by the number of items it recommends, and this is just one way to approximate reward. Average CTR is computed in every 2000 observations (not the accumulated CTR) for each algorithm based on the unbiased offline evaluation protocol proposed in [12, 31]. Following the same evaluation principle used in [12], we normalized the resulting CTR from different bandit algorithms by the corresponding logged random strategy's CTR. We report the normalized CTR results from different contextual bandit algorithms over 160 derived user groups in Figure 2.2 (a).

CoLin outperformed all baselines on this real-world dataset, except CLUB on the first day. Results from other user cluster sizes (40 and 80) showed consistent improvement as demonstrated in Figure 2.2 (a) with 160 user clusters; but due to space limit, we did not include those results. As we can find CLUB achieved the best CTR on the first day; but as some popular news articles became out-of-date, CLUB cannot correctly recognize their decreased popularity, and thus provided degenerated recommendations. But in CoLin, because of collaborative preference learning, it better controlled the exploration-exploitation trade-off and thus timely recognized the change of items' popularity. However, one potential limitation of CoLin is its computational complexity: because the dimension of global statistic matrix $\mathbf{A}_t$ defined in Eq (2.4) is $dN \times dN$, the running time of CoLin scales quadratically with the number of users. It makes CoLin less attractive in practical applications where the size of users is large. One potential solution is to enforce sparsity in the estimated $\mathbf{W}$ matrix such that distributed model update is possible, i.e., only share information within the connected users. The simulation study in Table 2.4 confirms the feasibility of this direction and we will explore it in our future work.

Experiments on LastFM & Delicious

The LastFM dataset is extracted from the music streaming service Last.fm, and the Delicious dataset is extracted the social bookmark sharing service website Delicious. These two datasets were generated by the Information Retrieval group at Universidad Autonomade Madrid for the HetRec 2011 workshop with the goal of investigating the usage of heterogeneous information in recommendation system[1]. The LastFM dataset contains 1,892 users and 17,632 items (artists). We used the information of "listened artists" of each user to create payoffs for bandit algorithms: if a user listened to an artist at least once, the payoff is 1, otherwise 0. The Delicious dataset contains 1,861 users and 69,226 items (URLs). We generated the payoffs using the information about the bookmarked URLs for each user: the payoff is 1 is the user bookmarked a particular

[1]Datasets and their full description is available at http://grouplens.org/datasets/hetrec-2011

URL, otherwise 0. Both of these two datasets contain the users' social network graph, which makes them a perfect real-world testbed for collaborative bandits.

Following the same settings in [11], we pre-processed these two datasets in order to fit them into the contextual bandit setting. We first used all tags associated with a single item to create a TF-IDF feature vector, which uniquely represents the item. Then we used PCA to reduce the dimensionality. In both datasets, we only retained the first 25 principle components to construct the context vectors, i.e., the feature dimension $d = 25$. We generated the candidate arm pool as follows: we fixed the size of candidate arm pool to be $K = 25$; for a particular user u, we picked one item from those nonzero payoff items for user u according to the whole observations in the dataset, and randomly picked the other 24 from those zero-payoff items for user u.

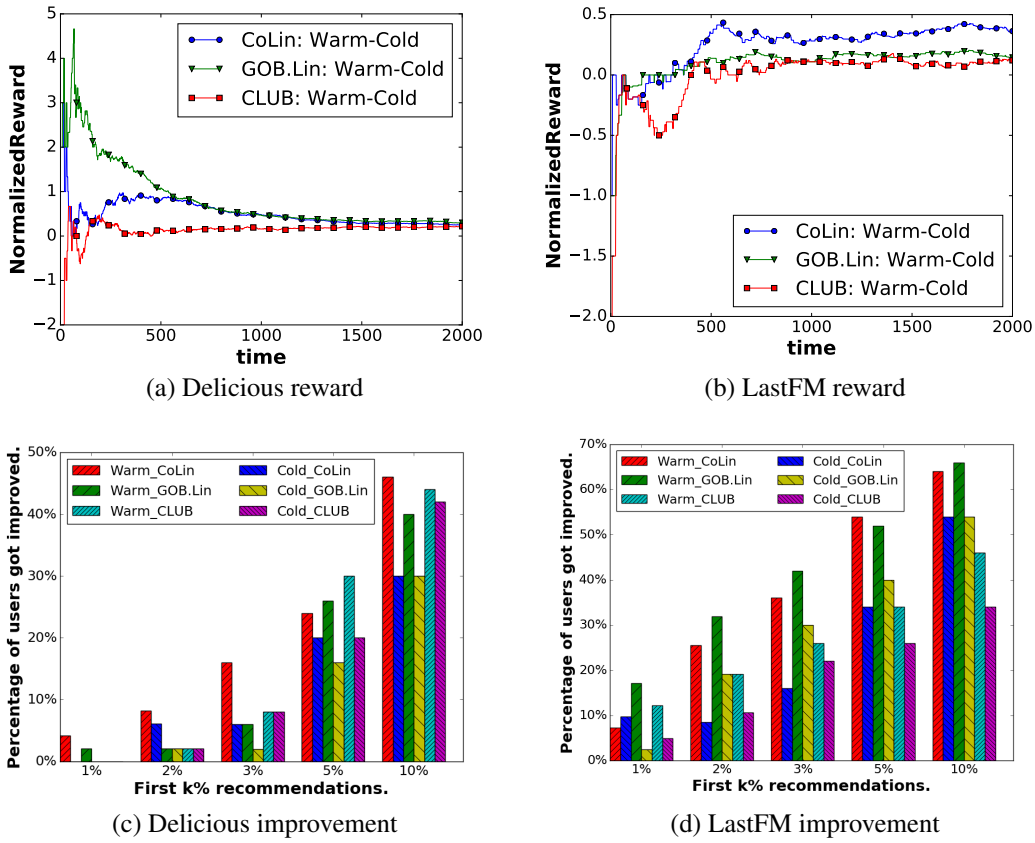
User relation graph is extracted from the social network provided by the datasets. In order to make the graph denser and make the algorithms computationally feasible, we performed graph-cut to cluster users into M clusters. Users in the same cluster are assumed to share the same bandit parameters. In our experiments, M was set to be 50, 100, and 200. Our reported results are from the setting of $M = 200$, and similar results were achieved in other settings of M. After user clustering, a weighted graph can be generated: the nodes are the clusters of nodes in the original graph; and the edges between different clusters are weighted by the number of inter-cluster edges in the original graph. In CoLin, we also need the diagonal elements in $\mathbf{W}$, which is undefined in a graph-cut based clustering algorithm. We computed the diagonal elements based on the derived regret bound of CoLin. Specifically, we first set $w_{ij} \propto c(i, j)$, where $c(i, j)$ is the number of edges between cluster i and j; then we optimized $\{w_{ii}\}_{i=1}^{N}$ which minimizes the term $\sum_i^N \sum_j^N w_{ij}^2$.

We included three variants of li2010contextual, hybrid li2010contextual, GOB.Lin and CLUB as baselines. The three variants of li2010contextual include: (1) li2010contextual that runs independently on each user, denoted as N-li2010contextual; (2) li2010contextual that is shared in each user cluster, denoted as M-li2010contextual (M is the number of clusters); (3) li2010contextual that is shared by all the users, denoted as Uniform-li2010contextual. Following the setting in [11], GOB.Lin also operates at the user cluster level and it takes the connectivity among clusters as input. We normalized the accumulated reward in each algorithm by a random strategy's accumulated reward, and compute the average accumulated normalized reward in every 50 iterations. Note that user features required by hybrid li2010contextual are not given in these two datasets. We applied the same strategy as we used in simulation to generate the user features.

From the results shown in Figure 2.2 (b) and (c), we can find that CoLin outperforms all the baselines on both Delicious and LastFM datasets. It is worth noting that these two datasets are structurally different, as shown in Figure 2.3 (a), the popularity of items on these two datasets differs significantly: on LastFM dataset, there are a lot more popular artists whom everybody listens to than the popular websites which everyone bookmarks on Delicious dataset. Thus the highly skewed distribution of item popularity makes recommendation on Delicious dataset much more challenging. Because most of items are only bookmarked by a handful of users, exploiting the relatedness among users to propagate feedback become vital. While on LastFM since there are much more popular items that most users would like, most algorithms can easily recognize the quality of items. In order to better understand this difference, we performed detailed item-level analysis to examine the effectiveness of different algorithms on items with varied popularity. Specifically, we first ranked all the items in these two datasets in a descending order of item popularity and then examined the item-based recommendation precision from all the bandit algorithms, e.g., percentage of item recommendations that are accepted by the users. In order to better visualize the results, we grouped the ranked items into different batches and report the average recommendation precision over each batch in Figure 2.3 (b) and Figure 2.3 (c).

From the results about the item-based recommendation precision, we can clearly find that on the LastFM dataset, CoLin achieved improved performance against all the baselines in every category of items, given the popularity of items in this dataset is more balanced. On Delicious dataset, CoLin achieved better performance on the top-ranked items; however, because of the skewness of item popularity, less popular items are still challenging for all the bandit algorithms to correctly recognize on this dataset.

This analysis motivates us to further analyze the user-level recommendation performance of different bandit algorithms, especially to understand the effectiveness of collaboration among users in alleviating the cold-start problem. To quantitatively evaluate this, we first ranked the user clusters in a descending order with respect to the number of observations in it. We then selected top 50 clusters as group 1. From the bottom 100 user



(a) Delicious reward (b) LastFM reward

(c) Delicious improvement (d) LastFM improvement

Figure 2.4: Effectiveness of collaboration and User-based analysis

clusters, we select 50 of them who are mostly connected to the users in group 1, and refer to them as group 2. The first group of users is called "learning bucket" and the second group is called "testing bucket." Based on this separation of user clusters, we performed two experiments: one is warm-start, and another is cold-start. In the warm-start setting, we first run all the algorithms on the learning bucket to estimate parameters for both group of users, such as $\mathbf{A}_t$ and $\mathbf{b}_t$ in CoLin. However, because users in the second group do not have any observation in the learning bucket, their model parameters can only be updated via the collaboration among bandits, i.e., in CoLin and GOBLin. Then with the model parameters estimated from the learning bucket, we evaluate different algorithms on the deployment bucket. Correspondingly, in the cold-start setting, we directly run and evaluate the bandit algorithms on the deployment bucket. It is obvious that since li2010contextual assumes users are independent and there is no information shared among users, li2010contextual's performance will not change under warm-start and cold start settings. While in CoLin, GOB.Lin and CLUB, because of the collaboration among users, information is propagated among users. In this case, user preference information learned from the learning bucket can be propagated to the deployment bucket.

We reported the performance on the first 10% observations in the deployment bucket instead of the whole observations, in order to better simulate the cold-start situation (i.e., all the algorithms do not have sufficient observations to confidently estimate model parameters). In Figure 2.4 (a) and (b), we reported the gap of accumulated rewards from CoLin GOB.Lin, and CLUB between warm-start and cold-start, normalized by rewards obtained from li2010contextual. From Figure 2.4(a) we can notice that on Delicious dataset, although at the very beginning of the warm-start setting both GOBLin and CoLin performed worse than the cold-start setting, both algorithms in warm-start quickly improved and outperformed the cold-start setting. One possible explanation is that the algorithms might take the first several iterations to adapt the models propagated from the first user group to the second. In particular, from Figure 2.4 (a), it is clear that once both algorithms are adapted, the improvement between warm-start and cold-start on CoLin is larger than that on GOB.Lin. This verified CoLin's effectiveness in address the cold-start challenge. From Figure 2.4 (b), we can find that warm-start helps both algorithms immediately at the first several iterations on LastFM dataset. This might be caused by the

flat distribution of item popularity in this dataset: users in the second group also prefer the items liked by users in the first group. We should note that the larger gap from GOB.Lin than that from CoLin between warm-start and cold-start settings does not mean CoLin is worse than GOB.Lin; but it indicates the cold-start CoLin learned faster than the cold-start GOB.Lin on this dataset. And the final performance of both cold-start and warm-start CoLin was better than GOB.Lin in the corresponding settings. We can also notice that cold-start CLUB performed very similarly as warm-start CLUB. It means the user clusters automatically constructed in CLUB does not help collaborative learning. This experiment confirms that appropriate observation sharing among users is vital to address the cold-start problem in recommendation.

Furthermore, we performed a user-based analysis to examine how many users will benefit from collaborative bandits. We define an improved user as the user who is served with improved recommendations from a collaborative bandit algorithm (i.e. CoLin, GOB.Lin and CLUB) than those from isolated li2010contextuals. We reported the percentage of improved users in the first 1%, 2%, 3%, 5%, and 10% observations during online learning. Figure 2.4 (c) and (d) demonstrate that in all collaborative bandit algorithms, the warm-start setting benefits much more users than cold-start setting. This further supports our motivation in developing bandit algorithms in a collaborative environment, which helps alleviate the cold-start challenge.

2.2 Exploration in Gradient Space with Structural Information

Online Learning to Rank (OL2R) [32] is a family of online learning solutions, which exploit implicit feedback from users to directly optimize parameterized rankers on the fly. It has drawn increasing attention in research community in recent years due to its advantages over classical offline learning to rank algorithms [33]. First, it avoids the expensive and time consuming process of offline result relevance annotation. Second, as it directly learns from user feedback, it optimizes the ranking results to best reflect current user preferences [34]. Third, because the model is updated on the fly, there is no need to store user click history offline, which alleviates many privacy concerns [17].

One strain of OL2R algorithms, represented by Dueling Bandit Gradient Descent (DBGD) [35], optimize a linear scoring function by exploring the parameter space via interleaved test. Algorithms of this type first propose an exploratory direction as a tentative model update direction, and then update the current ranker if the proposed direction provides better ranking utility. In practice, result utility is usually inferred from user clicks on an interleaved list of ranking results from each ranker [36]. The key technical insight of DBGD-type algorithms is that the expectation of selected directions is an *unbiased* estimate of true gradient of the unknown loss function for ranking [37]. As a result, DBGD is essentially a stochastic online gradient descent algorithm. However, because the exploration directions are uniformly sampled from the entire parameter space, when the dimensionality of the space is high (which is usually the case in practice), the *variance* in gradient estimation becomes large. This directly slows down the learning convergence of the algorithm and inevitably increases sample complexity.

Recently, several follow-up works have realized this deficiency of gradient exploration in DBGD, and propose various types of solutions to improve its learning efficiency. One type of studies explore multiple random directions in each iteration of model update. Unbiased estimate of gradient is maintained in this type of revisions of DBGD, as the directions are still uniformly sampled. Model estimation variance is expected to be reduced by testing more exploratory directions; but, in practice, as the users would only examine a finite number of documents under each query (e.g., due to position bias [38]), the sensitivity of interleaved test drops as a result of more exploratory rankers having to be tested at once. This unfortunately introduces additional variance in model estimation. Another type of research constrains the sampling space for gradient exploration [3, 39, 40]. However, this line of solutions cannot guarantee the estimated gradient remains unbiased, and thus face high risk of converging towards a sub-optimal solution.

Although empirically effective, previous OL2R solutions neglect an important property of click-based result utility evaluation: users only perceive utility from the documents that they actually examine. As a result, the *true* gradient is only revealed by features playing an essential role in ranking those examined documents under this query. Here we define essential features in ranking a particular set of documents as those features with non-zero variance among the documents. Assume in an interleaved test, one ranking feature takes a constant value in all examined documents under this query, such that it has no effect in differentiating the quality

of those documents. Then, the proposed exploratory direction's contribution to the ranker update on this particular dimension cannot be justified by this test result. Random gradient exploration hence introduces an arbitrary update on this dimension, which inevitably leads to high estimation variance over time. This example can be generalized to situations where multiple (even correlated) features have no effect in differentiating the utility of examined documents in the result of an interleaved test. Because in practice users usually only examine a handful of documents under each query [38, 41], but each document consists of hundreds or even thousands of ranking features, the variance introduced by random exploration on those non-essential features could be considerably large.

The above analysis suggests that an interleaved test only reveals the projection of true gradient in the spanned space of examined documents under a test query (termed the "document space" in this work). With this as our motivation, we decide to project the winning direction back into the document space so as to reduce the variance introduced by random gradient exploration. We construct the document space from inferred users' result examinations [41], which are not observable in the user response but can be statistically modeled. Because this projection is independent from how the proposal directions are created, this solution can be directly applied to any DBGD-type OL2R algorithm. We theoretically prove that the projected direction is still an unbiased estimate of the true gradient, i.e., model convergence is guaranteed, and also prove the reduced variance directly leads to considerable regret reduction in online model update. We compare the proposed method with several best-performing DBGD-type OL2R algorithms on a collection of large-scale learning to rank datasets and confirmed the effectiveness of our proposed solution.

2.2.1 Related Work

One key family of OL2R methods root in Dueling Bandit Gradient Descent (DBGD) [35], which uses online gradient descent to solve a bandit convex optimization problem [37]. In each iteration, DBGD uniformly samples a random direction from the entire parameter space to create an exploratory ranker, and uses an interleaved test [34] to compare the current ranker with the exploratory one. If the exploratory ranker is preferred, the proposed direction is used as the gradient to update the model. This procedure yields an unbiased estimate of true gradient [42]. However, the variance of DBGD's gradient estimation is high due to the nature of uniform exploration of the entire parameter space, which limits its learning efficiency.

Recently, attempts have been made to improve the learning efficiency of DBGD-type algorithms. Schuth et al. [43] proposed a Multileave Gradient Descent (MGD) algorithm to explore multiple stochastic directions in each iteration with multi-interleaving comparison [44]. Zhao and King [45] developed a Dual-Point Dueling Bandit Gradient Descent algorithm to sample two stochastic vectors with opposite directions as the candidate gradients. The basic idea of this line of solutions is to test more exploratory directions at once so as to obtain the true gradient estimate sooner. However, their gradient exploration is still within the entire feature space. As users often only examine a small number of documents under each query, the sensitivity of interleaved test drops due to more exploratory rankers need to be tested. In a different direction of solutions, researchers proposed to constrain the sampling space for gradient exploration. Hofmann et al. chose to filter the stochastic directions by historical comparisons before an interleaved test [39]. Oosterhuis et. al [40] proposed to explore gradients in a subspace constructed by a set of pre-selected reference documents from an offline training corpus. Wang et al. [3] proposed to use historical interactions to avoid repeatedly exploring less promising directions, which also reduces gradient exploration to a subspace. However, the variance of gradient exploration is reduced at a cost of introducing bias into gradient approximation, so that such algorithms have a risk of converging to sub-optimal results.

There are also other parallel lines of OL2R algorithms that do not explore the gradient space the way DBGD-type algorithms do, but directly optimize the ranking model from click feedback. Kveton et al. [46] proposed Cascading Bandits to learn from users' click behaviour, where skipped documents are assumed to be less attractive than later clicked ones. This model is then extended to the dependent click model [47] to support multiple clicks in one query, and further studied for general stochastic click models [48]. However, these algorithms estimate a separate model for each query and do not share estimation across queries, which lead to slow convergence. Oosterhuis et al. [49] proposed a Pairwise Differentiable Gradient Descent (PDGD) algorithm that constructs gradients from pairwise result comparisons to update the model, and can be used to optimize neural network models. We should note that our solution is not compatible nor directly comparable

with these non-DBGD algorithms, as there is no gradient exploration in these algorithms and our proposed gradient projection does not apply.

2.2.2 Document Space Projection for Online Learning to Rank

In this section we describe our proposed document space gradient projection method for online learning to rank. We first describe the problem setup in Section 2.2.2. Then we present Document Space Projected Dueling Bandit Gradient Descent (DBGD-DSP) algorithm as an example of our proposed general solution in Section 2.2.2. Our gradient projection method is independent from how the exploratory gradient is proposed, and thus can be directly applied to any existing DBGD-type OL2R algorithm [2] to reduce its variance of gradient estimation. We rigorously prove the unbiasedness of our gradient estimation in Section 2.2.2 and analyze the regret of DBGD-DSP in Section 2.2.2. The same procedure and conclusions can be applied to any DBGD-type algorithm of interest.

Problem Setup

The estimation of OL2R models can be formalized as a dueling bandit problem [35]. In iteration t, an OL2R algorithm receives a query and associated candidate documents, which are represented as a set of d-dimensional query-document pair feature vectors $X_t = \{x_1, x_2, ..., x_s\}$. The algorithm takes two actions: first, it proposes two rankers, whose parameters are denoted as w, w'; second, it ranks the given documents with these two rankers accordingly. An oracle (i.e., user) compares (duels) the two rankers' results and provides feedback. In practice, an interleaving method [34] is applied to merge the ranking lists of the two rankers and display the resulting ranked list to the user. User preference is inferred from the click feedback. Thus, the ranker that contributes more clicked documents is preferred. We denote $w \succ w'$ for the event that w is preferred over w'. The comparison between two individual rankers is determined independently of other comparisons performed before with a probability $P(w \succ w'|X_t)$, such that $P(w \succ w'|X_t) = P_t(w \succ w') = f_t(w, w')$. $f_t(w, w')$ can be viewed as the distinguishability of the two rankers w and w' by an interleave comparison under query X_t.

We quantify the performance of an online learning algorithm using cumulative regret defined as follows:

$$R(T) = \sum_{t=1}^{T} f_t(w^*, w_t) + f_t(w^*, w'_t), \tag{2.9}$$

where w_t and w'_t are rankers compared at time t, and w^* is the best ranker in ground-truth. As a result, the distinguishability measure $f_t(w^*, w)$ indicates the loss of proposing a sub-optimal ranker w. We denote $f_t(w_t, w)$ as $f_t(w)$ for simplicity. The goal of an OL2R algorithm is to optimize its parameter towards w^* according to loss $f_t(w)$. A desired OL2R algorithm should have a sublinear regret in a finitie time horizon T, so that the one-step regret is quickly decreasing to zero over time.

In this work, we make the following assumptions similar to [35]. We assume an unknown utility function $v_t(w)$ that quantifies the quality of a ranker w over query X_t. The utility function v_t is assumed to be differentiable, strongly concave and L_v-Lipschitz, which means $|v_t(x) - v_t(y)| \leq L_v|x - y|$.

A link function σ describes the probabilistic comparison of utilities of two rankers as,

$$P_t(w \succ w') = f_t(w, w') = \sigma(v_t(w) - v_t(w')).$$

The link function should be rotation-symmetric, which means $\sigma(x) = 1 - \sigma(-x)$. We assume the link function is L_σ-Lipschitz and second order L_2-Lipschitz. The link function behaves like a cumulative probability distribution function. For example, a common choice of link function is the standard logistic function $\sigma(x) = \frac{1}{1+\exp(-x)}$, which satisfies all the assumptions.

Algorithm 2 Document Space Projected Dueling Bandit Gradient Descent (DBGD-DSP)

Inputs: δ, α
Initiate $w_1 = \textit{sample\_unit\_vector}()$
for $t = 1$ to T **do**
  Receive query $X_t = \{x_1, x_2, ..., x_s\}$
  $u_t = \textit{sample\_unit\_vector}()$
  $w'_t = w_t + \delta u_t$
  Generate ranked lists $l(X_t, w_t), l(X_t, w'_t)$
  Set $L_t = \text{Interleave}\big(\{l(X_t, w_t), l(X_t, w'_t)\}\big)$, and present L_t to user
  Receive click positions C_t on L_t, and infer click credits $\{c_t, c'_t\}$
  if $c_t \geq c'_t$ **then**
    $w_{t+1} = w_t$
  else
    Based on C_t, infer user examined top m_t documents in L_t.
    Solve the orthogonal projection matrix $\mathbf{A}_t$ for document space $S_t = span(\{x_{L_t,1}, x_{L_t,2}, ..., x_{L_t,m_t}\})$.
    Project u_t onto S_t by $g_t = \mathbf{A}_t u_t$
    $w_{t+1} = w_t + \alpha g_t$

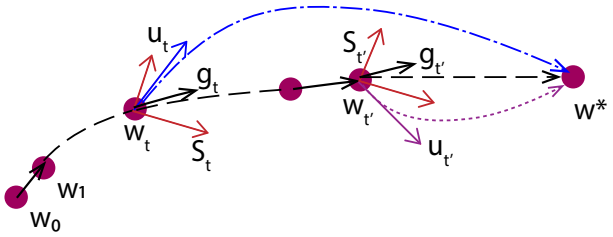


Figure 2.5: Illustration of model update for DBGD-DSP in a three dimensional space. Dashed lines represent the trajectory of DBGD following different update directions. u_t is the selected direction by DBGD, which is in the 3-d space. Red bases present the document space S_t on a 2-d plane. u_t is projected onto S_t to become g_t for model update.

Document Space Projected Dueling Bandit Gradient Descent

We describe our proposed Document Space Projected Dueling Bandit Gradient Descent (DBGD-DSP) in Algorithm 2. We should note it fits all DBGD-type OL2R algorithm settings. At the beginning of iteration t, user initiates a query X_t. We denote w_t as the parameter of the current ranker. DBGD-DSP first uniformly samples a vector u_t from d dimensional unit sphere $\mathbb{S}^{d-1}$ (i.e., $|u_t|_2 = 1$) as an exploratory direction, and proposes a candidate ranker $w'_t = w_t + \delta u_t$, where δ is the step size of exploration. The algorithm then uses the two rankers (w_t and w'_t) to generate ranking lists $l(X_t, w_t)$ and $l(X_t, w'_t)$ accordingly, and combines them with an interleaving method, such as Team Draft Interleaving [34] or Probabilistic Interleaving [50]. The user examines the result list and provides implicit click feedback to indicate their relevance evaluation of the results. The interleaving method uses this implicit feedback to infer which ranker is preferred by the user. If the exploratory ranker is preferred (i.e., wins the duel), previous DBGD-style algorithms update the current ranker by $w_{t+1} = w_t + \alpha u_t$, where α is the learning rate; otherwise the current ranker stays intact. This gradient exploration strategy yields an unbiased estimate of the true gradient [37], in terms of expectation.

However, since the exploratory gradient u_t is required to be uniformly sampled from the entire d dimensional unit sphere $\mathbb{S}^{d-1}$, the model update suffers from high variance in its gradient estimation, especially when d is large, as in practice. Various improvements to this issue have been proposed in the past, but they still introduce other difficulties, such as variance and bias trade-off [3, 39, 40], and test sensitivity and efficiency [44, 45].

[2] In the following discussions, we will use "DBGD-type OL2R algorithm" and "OL2R algorithm" interchangeably, as the focus of this work is improving the efficiency of DBGD-type OL2R algorithms.

Unlike previous works that reduce the sampling space of gradient exploration before the interleaved test [3,39,40], we change the winning direction after the test. The key insight is that only the projected true gradient in the spanned space of *examined* documents under query X_t (denoted as document space S_t) can be revealed by an interleaved test. For example, as shown in Figure 2.5, a DBGD-style algorithm is comparing the current ranker w_t and $w'_t = w_t + \delta u_t$ with a uniformly sampled exploration direction u_t. The user examines top m documents, e.g., $\{x_1, ..x_m\}$, of the interleaved ranking list (of course m is unknown to the algorithm) and w'_t wins the duel. The estimated gradient u_t can therefore be separated into two components, one component g_t that belongs to the document space $S_t = span\{x_1, ..x_m\}$ and the other component $u_t - g_t$ that is orthogonal to document space S_t. The orthogonal component $u_t - g_t$ does not affect the ranking among the examined documents, i.e. $(w_t + \delta u_t)^T x_i = (w_t + \delta g_t)^T x_i$, and thus does not contribute to the loss function and true gradient estimation. Intuitively, $u_t - g_t$ is not supported by the observed interleaved test, as anything sampled from the complement of S_t cannot be verified by the examined documents. As a result, it is safe to exclude the direction $u_t - g_t$ from model update, which we later prove maintains the unbiasedness of the original DBGD-type gradient estimation, and reduces the variance. As illustrated in Figure 2.5, although u_t will eventually lead to the same model estimation, as it is unbiased, this guarantee is only obtained in expectation. The variance could potentially be large: for example, the blue and purple updating traces slow down model convergence, when the number of observations is finite.

As shown in line 14 to 16 of Algorithm 2, we solve for the orthogonal projection matrix $\mathbf{A}_t$ of document space S_t, and project the selected direction u_t onto the document space S_t after each interleaved test. We leave the detailed design of constructing document space and solving projection matrix $\mathbf{A}_t$ in Section 2.2.2. Before that, we first rigorously prove the projection maintains an unbiased estimate of true gradient in Section 2.2.2. Since the document space is constructed only by the examined documents, the rank of document space is expected to be smaller than the entire parameter space. This directly leads to lower variance and faster model convergence. We show that our document space projection reduces the variance of gradient estimation from d to $Rank(\mathbf{A}_t)$ in Section 2.2.2, and then analyze its benefit for regret reduction from a low-variance gradient estimation.

Unbiasedness of Gradient Estimation

We now prove that our document space projected gradient is an unbiased estimate of true gradient in the sense of expectation [35]. We define $Z_t(w)$ as the event of w winning the duel with w_t,

$$Z_t(w) = \begin{cases} 1 & \text{w.p. } 1 - P_t(w_t \succ w) \\ 0 & \text{w.p. } P_t(w_t \succ w) \end{cases}$$

Then the gradient used for model update in DBGD-DSP (as described in Algorithm 2) can be described as,

$$h_t = -Z_t(w_t + \delta u_t) g_t. \tag{2.10}$$

Note that by adding a negative sign we view our model update as online gradient descent $w_{t+1} = w_t - \alpha g_t$.

We now show in the following theorem that this is an unbiased gradient estimation of true gradient. By defining a smoothed version of f_t as $\hat{f}_t(w) = \mathbb{E}_{u \in \mathbb{B}}[f_t(w + \delta u)]$, we have:

Theorem 2. *The projected gradient g_t in DBGD-DSP is an unbiased estimate of true gradient, i.e.,*

$$\mathbb{E}[h_t] = \frac{\delta}{d} \nabla \hat{f}_t(w) \tag{2.11}$$

over random unit vector u_t.

Proof. Based on the Lemma 1 of [35], we have

$$\mathbb{E}\left[h_t\right] = \mathbb{E}\left[-Z_t(w_t + \delta u_t)\mathbf{A}_t u_t\right] = \mathbb{E}_{u_t \in \mathbb{S}^{d-1}}\left[f_t(w + \delta \mathbf{A}_t u_t) u_t\right]$$

Define $F_t(w) = f_t(\mathbf{A}_t w)$, we have

$$\begin{aligned}
\mathbb{E}[h_t] &= \mathbb{E}_{u_t \in \mathbb{S}^{d-1}} \left[f_t(w_t + \delta \mathbf{A}_t u_t) u_t \right] \\
&= \mathbb{E}_{u_t \in \mathbb{S}^{d-1}} [F_t(\mathbf{A}_t^{-1} w_t + \delta u_t) u_t] \\
&= \frac{\delta}{d} \nabla \mathbb{E}_{u_t \in \mathbb{B}^d} [F_t(\mathbf{A}_t^{-1} w_t + \delta u_t)] \\
&= \frac{\delta}{d} \nabla \hat{F}_t(\mathbf{A}_t^{-1} w_t) \\
&= \frac{\delta}{d} \mathbf{A}_t \nabla \hat{f}_t(w_t) \\
&= \frac{\delta}{d} \nabla \hat{f}_t(w_t)
\end{aligned}$$

where the third equality is based on Stokes' Theorem. The last equality holds because gradient $\nabla \hat{f}_t(w_t)$ belongs to document space S_t, and thus projecting it by $\mathbf{A}_t$ maps back to itself. □

The guarantee of unbiased gradient estimation is a major advantage of our proposed document space gradient projection method, compared with previous attempts to reduce the gradient exploration space, such as Oosterhuis et. al [40] and Wang et al. [3]. Our method enjoys reduced variance of gradient estimate (which will be proved next), without the risk of converging towards a sub-optimal solution. We should note that the above is independent from the mechanism of how the proposal directions are generated, as shown in the first four steps of proof above. As a result, if the input direction to our projection procedure is unbiased, the resulting update direction is also unbiased. This enables our solution's generalization to other types of DBGD algorithms.

Regret Analysis of DBGD-DSP

We now analyze the regret of our proposed DBGD-DSP algorithm, starting with its variance of gradient update.

Lemma 2. *The variance of gradient update in DBGD-DSP is bounded by*

$$\mathbb{E}[|h_t|^2] = \mathbb{E}_{u_t \in \mathbb{S}^{d-1}} \left[| - Z_t(w_t + \delta u_t) \mathbf{A}_t u_t|^2 \right] \leq \frac{Rank(\mathbf{A}_t)}{d}.$$

Proof.

$$\begin{aligned}
\mathbb{E}[|h_t|^2] &= \mathbb{E}_{u_t} \left[| - Z_t(w_t + \delta u_t) \mathbf{A}_t u_t|^2 \right] \\
&\leq \mathbb{E}_{u_t} \left[|\mathbf{A}_t u_t|^2 \right] \\
&= \mathbb{E}_{u_t} \left[(\mathbf{A}_t u_t)^\top (\mathbf{A}_t u_t) \right] \\
&= \mathrm{tr} \left(\mathbb{E}_{u_t} \left[\mathbf{A}_t u_t u_t^\top \mathbf{A}_t^\top \right] \right) \text{ //apply the trace trick} \\
&= \mathrm{tr} \left(\mathbf{A}_t \mathbb{E}_{u_t} \left[u_t u_t^\top \right] \mathbf{A}_t^\top \right) \\
&= \mathrm{tr} \left(\mathbf{A}_t \frac{1}{d} I \mathbf{A}_t^\top \right) \\
&= \frac{1}{d} \mathrm{tr} \left(\mathbf{A}_t \mathbf{A}_t^\top \right) \\
&= \frac{1}{d} \mathrm{tr} \left(\mathbf{A}_t \right) \text{ //a projection matrix is idempotent} \\
&= \frac{\mathrm{Rank}(\mathbf{A}_t)}{d}
\end{aligned}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace operation. The sixth equality holds because u_t is uniformly sampled from a unit sphere, and its covariance matrix $\mathbb{E}_{u_t}\left[u_t u_t^\top\right]$ is $\frac{1}{d}I$. Since $\mathbf{A}_t$ is an orthogonal projection matrix, the eighth equality holds for $\mathbf{A}_t\mathbf{A}_t^\top = \mathbf{A}_t$. □

Remark 1. *The variance of gradient update in DBGD [35] is bounded by* $\mathbb{E}_{u_t}\left[|-Z_t(w_t+\delta u_t)u_t|^2\right] \leq 1$.

Comparing the variance of gradient update in DBGD-DSP with DBGD, our method reduces the variance from 1 to $\frac{\mathrm{Rank}(\mathbf{A}_t)}{d}$. Since the dimension of projection matrix $\mathbf{A}_t$ is d-by-d, we have $\mathrm{Rank}(\mathbf{A}_t) \leq d$, which guarantees the reduction of variance in DBGD-DSP comparing to that in DBGD. The rank of $\mathbf{A}_t$ is also bounded by the number of *examined* documents m_t, since document space S_t is constructed by these m_t examined documents. In practice, users would only examine a handful of documents [38, 41], while the ranking feature dimension is expected to be much larger. We argue that $m_t \ll d$, such that our document space projection achieves considerable variance reduction.

The significance of this variance reduction can be intuitively understood from Figure 2.5: though different traces of model update would eventually lead to the same converged model, if one has a sufficiently large amount of interactions with users, the one with lower variance would always require less observations. A faster converging algorithm leads to user satisfaction earlier. Next, we verify this benefit by proving the reduction of regret introduced by the reduced variance in gradient estimation.

Theorem 3. *By setting*

$$m = \max_t m_t, \delta = \frac{\sqrt{2Rm}}{\sqrt{13L}T^{1/4}}, \alpha = \frac{Rm}{\sqrt{T}\delta},$$

the expected regret of DBGD-DSP as defined in Eq (2.9) *is upper bounded by,*

$$\mathbb{E}[Reg] \leq 2\lambda_T T^{3/4}\sqrt{26RmL}, \tag{2.12}$$

where

$$\lambda_T = \frac{L_\sigma\sqrt{13L}T^{1/4}}{L_\sigma\sqrt{13L}T^{1/4} - L_v L_2\sqrt{2Rm}}$$

The proof is obtained by extending Theorem 2 in [35]. We omit the details due to space limit, and emphasize that the key difference is introduced by replacing variance of gradient estimation from $\mathbb{E}_{u_t}\left[|-Z_t(w_t+\delta u_t)u_t|^2\right]$ to $\mathbb{E}_{u_t}\left[|-Z_t(w_t+\delta u_t)\mathbf{A}_t u_t|^2\right]$. Since the variance of gradient estimation is reduced from 1 to $\frac{\mathrm{Rank}(\mathbf{A}_t)}{d}$, the regret of DBGD can be reduced from $O(\sqrt{d}T^{3/4})$ to $O(\sqrt{m}T^{3/4})$, where m is the maximum number of documents included in a document space under a single query. Again, as the number of included ranking features is oftentimes much larger than the number of documents a user would examine under a single query, the reduction of regret is considerable. Moreover, as the reduction of variance from our project-based method is independent from the way about how the proposal directions are generated, our method can be generally applied to most existing DBGD-type OL2R algorithms to improve their learning convergence.

Practical Treatments of Document Space Projection

Now we discuss several practical treatments of our proposed Document Space Projection method, including the construction of document space and orthogonal projection matrix.

In our theoretical analysis, we have assumed the knowledge of users' examined documents and corresponding projection matrix. However, in practice, a user's result examination is unobserved. A rich body of research has been developed to perform statistical inference of it, collectively known as click modeling [41, 51]. Any of these existing click models can be plugged into our solution framework, i.e., line 13 of Algorithm 2. In this work, we simply follow [38] to infer user examination by the last clicked position: given the click position list C_t, we use the last clicked position $c_{l,t}$ to approximate the last examined position M_t by setting $M_t = c_{l,t} + k$, where k is a hyper-parameter. Based on sequential examination hypothesis of click modeling, every document before the last clicked position is examined, and we use k to approximate the number of positions following

the last clicked position that were still examined. We leave more comprehensive study of click modeling in our solution as future work.

The above treatment provides a reasonable inference of examined documents. However, it requires a careful choice of k for each query (preferably). If k is set too large, the variance of gradient estimate will increase (as proved in Lemma 2). If k is too small, the document space may not include all examined documents, and it is at risk of introducing bias in gradient projection. To avoid bias in constructing the document space, we also consider adding historically examined documents to the current query's document space. Specifically, we add r recently examined documents to the current document space S_t to compensate the potentially overlooked examined documents in the current query.

In line 14 of Algorithm 2, we solve the orthogonal projection matrix $\mathbf{A}_t$ of document space S_t. $\mathbf{A}_t$ could be computed by several methods. Denote D_t as a d-by-m_t matrix where each column is the feature vector for an examined document. One can use QR decomposition or Singular Value Decomposition (SVD) to solve for its orthonormal basis V_t, and the projection matrix can then be constructed by $\mathbf{A}_t = V_t V_t^T$. In our experiments, we chose SVD for constructing the basis of document space, because of its widely available and efficient large-scale implementations. But the choice for the construction of this project matrix does not affect the convergence nor unbiasedness of our proposed solution.

2.2.3 Experiments

To demonstrate our proposed Document Space Projection method's empirical efficacy, we compare the performance of several best-performing DBGD-type OL2R algorithms on five public learning to rank datasets, with and without our document space projection method applied.

Experiment Setup

• **Datasets.** We tested our algorithms and the baselines on five benchmark datasets: including MQ2007, NP2003 [52], MSLR-WEB10K [53], and the Yahoo! Learning to Rank Challenge dataset [54]. In each of the five datasets, each query-document pair is encoded as a vector of ranking features. These features include PageRank, TF.IDF, Okapi-BM25, URL length, language model score, and many more varied by dataset.

The MQ2007 dataset is collected from the 2007 Million Query track at TREC [55]. MQ2007 contains about 1700 queries, which represent a mix of informational and navigational search intents. They both have 46-dimensional feature vectors to represent query-document pairs, and the document relevance are labeled in three grades: 0 (not relevant), 1 (relevant), and 2 (most relevant).

The NP2003 dataset also comes from the TREC Web track, consisting of queries crawled from the .gov domain. It is comprised of about 150 navigational-focused queries, with over 1000 document relevance assessments per query. It uses 64 ranking features, and the document relevance labels are binary (0 and 1 only).

The MSLR-WEB10K dataset was released by Microsoft in 2010, and consists of 10,000 queries with relevance assessments coming from a labeling set from the Microsoft Bing search engine. It has 136 ranking features, and the relevance judgments range from 0 (not relevant) to 4 (most relevant).

The Yahoo! Learning to Rank Challenge dataset was also released in 2010, as an effort on part of Yahoo! to promote the dataset as well as research into better learning to rank algorithms. The dataset contains about 36,000 queries, 883,000 assessed documents, and 700 ranking features. Again, the relevance judgments range from 0 (not relevant) to 4 (most relevant)

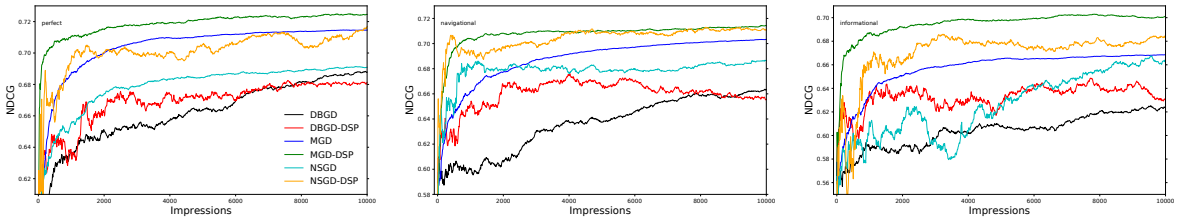
This diversity in the structure of the datasets that we chose to test on helps us to evaluate our algorithms more holistically. While small, the MQ2007 sets have been around for a long time and have a good mix of query types. NP2003 gives us insight into how the algorithms perform on navigational search intents specifically, which are markedly different in nature from informational search intents. MSLR-WEB10K and the Yahoo! dataset are large-scale datasets used by actual commercial search engines, which give us a better understanding of how the algorithms perform in practice. Since each dataset was split into training, testing, and validation

Table 2.5: Configurations of simulation click models.

	Click Probability					Stop Probability				
R	0	1	2	3	4	0	1	2	3	4
Per	0.0	0.2	0.4	0.8	1.0	0.0	0.0	0.0	0.0	0.0
Nav	0.05	0.3	0.5	0.7	0.95	0.2	0.3	0.5	0.7	0.9
Inf	0.4	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4	0.5

subsets, we used the training sets for online experiments to measure cumulative performance, and used the testing sets for evaluating offline performance.

• **Simulated User Interactions.** Based on an online learning to rank framework proposed in [49], we use the standard setup to simulate user interactions. Within this framework, we used the Cascade Click Model to simulate user click behavior. This model assumes that a user interacts with a set of search results by linearly scanning the list from top and making a decision for each document as to whether or not to click. In the model, the probability of a click for a given document is conditioned on the relevance label of that document, as a user is expected to be more likely to click on relevant documents. After evaluating each document, the user must decide whether or not to continue perusing the list. This decision's probability distribution is again conditioned on the relevance of the examined document, as a user is more likely to stop looking through the results if he/she has already satisfied their information need. These aforementioned probabilities can be altered to simulate different types of users and interactions.



(a) Perfect (b) Navigational (c) Informational

Figure 2.6: Offline NDCG@10 on Yahoo! dataset.

As illustrated in Table 2.5, we use three different click model probability configurations to represent three different types of users. First, we have the *perfect* user, who clicks on all relevant documents and does not stop browsing until they have visited all of the documents. This type of users contribute the least noise, as they make no mistakes and the feedback is entirely accurate. Second, we have the *navigational* user, who is very likely to click on the first highly relevant document that he/she sees and stops there. Third, we have the *informational* user, who, in his/her search for information, sometimes clicks on irrelevant documents, and as such contributes a significant amount of noise in click feedback.

• **Evaluation Metrics.** As set forth in [56], cumulative (online) Normalized Discounted Cumulative Gain (NDCG) and offline NDCG are commonly used metrics for evaluating OL2R algorithms. Cumulative NDCG is calculated by summing NDCG scores from successive iterations with a discount factor γ set to 0.995. We assess our model's estimation convergence via cosine similarity between the current weight vector and a reference weight vector (considered to be the optimal vector) as estimated by an offline learning-to-rank algorithm trained with the complete true relevance judgment labels. Due to its superior empirical performance, we used LambdaRank [57] with no hidden layer in our experiments to estimate this reference weight vector. In each experiment, the number of iterations T was set to 10,000, and the current query X_t was randomly sampled from the dataset in each iteration. We execute all the experiments 15 times with different random seeds, and report and compare the average performance in all experiments.

• **Evaluation Questions.** To better understand the advantages of our proposed algorithms, we aim to answer the following evaluation questions through the course of our experiments.

Q1: Can our proposed Document Space Projection method consistently improve the performance of best-performing DBGD-type OL2R algorithms?

Q2: Do gradients rectified by our document space projection explore the gradient space more efficiently?

Q3: How do different hyper-parameter settings alter the performance of our document space projection?

• **Baseline Algorithms.** We choose the following three best-performing DBGD-type OL2R algorithms as our baselines for comparison:

- **DBGD** [35]: A single direction uniformly sampled from the whole parameter space is explored.
- **MGD** [43]: Multiple directions are explored in one iteration to reduce the gradient estimation variance. Multileaving is used to compare multiple rankers. The model updates towards the mean of all rankers that beat the current model.
- **NSGD** [3]: Multiple directions are sampled from the null space of previously poorly performing gradients. Ties are broken by evaluating the tied candidate rankers on a recent set of difficult queries.

We apply our proposed Document Space Projection to the baseline algorithms, and compare them with DBGD-DSP, MGD-DSP and NSGD-DSP, respectively.

Table 2.6: Online NDCG@10, standard deviation and relative improvement of document space projection of each algorithm after 10,000 queries.

Click Model	Algorithm	**MQ2007**	**MSLR-WEB10K**	**NP2003**	**Yahoo**
Perfect	DBGD	679.3 (21.6)	532.2 (15.3)	1130.2 (43.3)	1165.5 (22.6)
	DBGD-DSP	689.1 (19.5)(+1.44%)	553.6 (13.1)(+4.02%)	1198.8 (40.0) (+6.07%)	1198.8 (33.5)(+2.86%)
	MGD	689.1 (14.6)	558.3 (7.0)	1192.9 (44.6)	1201.9 (16.3)
	MGD-DSP	**757.3 (16.2)(+9.90%)**	626.4 (9.6)(+12.20%)	1335.3 (39.1)(+11.94%)	**1309.4 (10.6) (+8.94%)**
	NSGD	684.4 (20.5)	589.5 (14.2)	1274.9 (47.4)	1162.3 (12.9)
	NSGD-DSP	732.5 (20.0)(+7.03%)	**635.6 (12.8)(+7.82%)**	**1368.5 (41.1)(+7.34%)**	1270.1 (2.5)(+9.27%)
Navigational	DBGD	646.1 (23.4)	517.5 (20.9)	1062.3 (55.4)	1133.3 (40.8)
	DBGD-DSP	664.9 (26.9)(+2.91%)	543.1 (14.8)(+4.95%)	1140.1 (52.5)(+7.32%)	1199.4 (34.6)(+5.83%)
	MGD	632.7 (15.5)	538.2 (7.2)	1115.4 (44.6)	1171.3 (20.4)
	MGD-DSP	**694.5 (15.7)(+9.77%)**	586.9 (9.5)(+9.05%)	**1300.9 (39.6)(+16.63%)**	**1290.2 (15.3) (+10.15%)**
	NSGD	660.1 (24.5)	562.1 (18.8)	1211.1 (66.5)	1186.2 (16.8)
	NSGD-DSP	724.6 (24.5)(+9.77%)	**608.3 (12.1) (+8.22%)**	1296.2 (24.3) (+7.03%)	1283.4 (7.2)(+8.19%)
Informational	DBGD	583.4 (46.0)	472.4 (34.6)	849.8 (144.5)	1107.3 (46.6)
	DBGD-DSP	620.1 (40.8)(+6.29%)	522.1 (18.6) (+10.52%)	992.5 (81.1)(+16.79%)	1158.5 (22.0)(+4.62%)
	MGD	621.2 (18.2)	538.3 (10.8)	1107.9 (46.2)	1146.6 (37.5)
	MGD-DSP	**671.4 (18.9)(+8.08%)**	580.5 (10.4)(+7.84%)	**1274.5 (42.9)(+15.04%)**	**1268.1 (16.4)(+10.60%)**
	NSGD	629.7 (25.3)	532.9 (15.2)	1123.5 (59.8)	1110.5 (10.9)
	NSGD-DSP	703.6 (29.2)(+11.74%)	**597.9 (14.1)(+12.20%)**	1222.8 (43.8)(+9.03%)	1204.7 (9.6)(+8.48%)

Performance of Document Space Projection

We begin our experimental analysis by answering our first evaluation question. We compared all algorithms over 3 click models and 5 datasets. We set the hyper-parameters of DBGD, MGD and NSGD according to their original papers. Following [35,43], we set the exploration step size δ to 1 and learning rate α to 0.1. Both MGD and NSGD explore 9 proposal directions in one iteration. For our document space projection method, we consider $k = 3$ documents following the last clicked position as examined documents, and add $r = 10$ recently examined documents into document space S_t. We use SVD to solve for orthonormal basis V_t of the document space S_t, and compute the projection matrix by $A_t = V_t V_t^\top$.

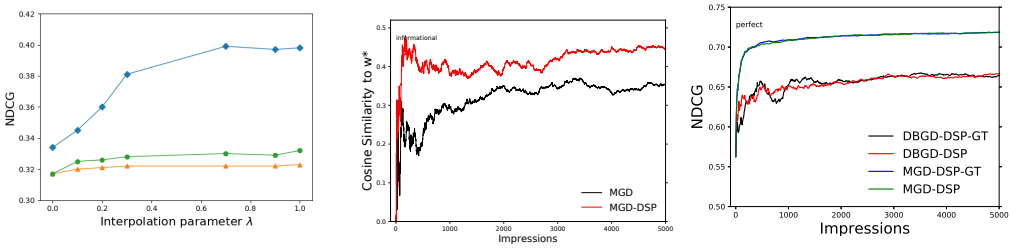
We reported the offline NDCG@10 and online cumulative NDCG @10 after 10,000 iterations in Table 2.6 and Table 2.7. Due to space limit, we only reported the offline performance during the 10,000 iterations over 3 click models on Yahoo dataset, a large-scale real-world L2R dataset with 700 ranking features, in Figure 2.6. MGD improves the online performance over DBGD by exploring multiple rankers simultaneously, and NSGD further improves over MGD by exploring gradients in a constrained subspace, as shown in Table 2.6. We

Table 2.7: Offline NDCG@10, standard deviation and relative improvement of document space projection of each algorithm after 10,000 queries.

Click Model	Algorithm	**MQ2007**	**MSLR-WEB10K**	**NP2003**	**Yahoo**
Perfect	DBGD	0.484 (0.023)	0.331 (0.009)	0.737 (0.056)	0.688 (0.011)
	DBGD-DSP	0.480 (0.020) (-0.83%)	0.333 (0.011) (+0.6%)	0.738 (0.059) (+0.14%)	0.681 (0.013) (-1.02%)
	MGD	0.495 (0.022)	0.334 (0.003)	0.746 (0.048)	0.715 (0.002)
	MGD-DSP	**0.501 (0.021)(+1.21%)**	**0.409 (0.006)(+22.46%)**	0.748 (0.055)(+0.27%)	**0.725 (0.003)(+1.40%)**
	NSGD	0.488 (0.019)	0.397 (0.012)	0.743 (0.050)	0.691 (0.005)
	NSGD-DSP	0.491 (0.022)(+0.61%)	**0.398 (0.008) (+0.25%)**	**0.750 (0.042) (+0.94%)**	0.717 (0.004)(+3.76%)
Navigational	DBGD	0.463 (0.028)	0.320 (0.012)	0.728 (0.054)	0.663 (0.020)
	DBGD-DSP	0.465 (0.024)(+0.43%)	0.327 (0.011)(+2.19%)	0.734 (0.052)(+0.82%)	0.656 (0.013)(-1.06%)
	MGD	0.426 (0.019)	0.321 (0.003)	0.740 (0.048)	0.703 (0.010)
	MGD-DSP	0.467 (0.021)(+9.62%)	0.331 (0.005)(+3.12%)	0.744 (0.053)(+0.54%)	**0.714 (0.006)(+1.56%)**
	NSGD	0.473 (0.022)	**0.389 (0.013)**	0.732 (0.053)	0.686 (0.008)
	NSGD-DSP	**0.478 (0.020)(+1.06%)**	0.376 (0.014)(-3.34%)	**0.788 (0.006)(+7.65%)**	**0.711 (0.001)(+3.64%)**
Informational	DBGD	0.410 (0.034)	0.294 (0.022)	0.699 (0.063)	0.623 (0.037)
	DBGD-DSP	0.427 (0.027)(+4.15%)	0.309 (0.011)(+32.65%)	0.692 (0.062)(-1.00%)	0.63 (0.030)(1.12%)
	MGD	0.406 (0.020)	0.317 (0.003)	0.726 (0.050)	0.668 (0.044)
	MGD-DSP	0.444 (0.025)(+0.44%)	0.325 (0.004)(+0.33%)	0.738 (0.054)(+0.74%)	**0.701 (0.005)(+4.94%)**
	NSGD	**0.469 (0.018)**	**0.360 (0.013)**	0.733 (0.056)	0.663 (0.015)
	NSGD-DSP	0.466 (0.019)(-0.64%)	0.340 (0.018)(-5.56%)	**0.789 (0.013)(+7.64%)**	0.685 (0.004)(+3.32%)

observe that our proposed document space projection method consistently improves the online performance of all baseline algorithms. Recall that in Section 2.2.2 our theoretical analysis suggested that document space projection reduces both the gradient estimation variance and the regret (online performance) with respect to the ratio between the rank of document space and feature dimension. Correspondingly, we observe that indeed we improved the OL2R models' ranking performance significantly over MSLR-WEB10K and Yahoo datasets, which are collected from real-world commerical search engines and have much higher feature dimensions (130 and 700 respectively). This result demonstrates the potential of document space projection to improve large-scale real-world DBGD-type OL2R applications with high-dimensional ranking features, as our algorithm attains satisfactory performance earlier than other baseline OL2R algorithms measured by online NDCG@10. We also notice that the standard deviation of those models' ranking performance is reduced when applying document space projection, which confirms our analysis of variance reduction in Lemma 2.

From Figure 2.6 and Table 2.7 we notice that document space projection mostly improves offline performance over baseline algorithms. Figure 2.6 shows that document space projection significantly accelerates the convergence rate over the baseline algorithms, because of the reduced variance in gradient estimation. We also observe that applying document space projection under the perfect click model may lead to degraded performance, for example DBGD on MQ2007 and Yahoo dataset. This is because document space projection guarantees an unbiased gradient estimation under the assumption of known result examinations, as discussed in Section 2.2.2. However, since in practice a user's result examination is unobserved, we approximated the examined documents by including all documents before the last clicked position and k additional documents after the last clicked position. The perfect click model is an ideal case that users' stop probability is set to 0.0 (see Table 2.5) and every document is examined. Here, the document space needs to include all displayed documents to guarantee the unbiasedness, which requires a significantly larger k compared to the k used for navigational and informational click models. We argue that in practice since users only examine a handful of documents, we could well-approximate the examined documents with a reasonable choice of k. More sophisticated click models can also be introduced. We will analyze the effect of k in Section 2.2.3. In addition, we also observe that under informational click model the performance of NSGD-DSP is slightly decreased compared with original NSGD over three datasets. Note that since NSGD does not guarantee its gradient exploration is unbiased, further projecting its gradient may also lead to a biased gradient update and thus a sub-optimal model.

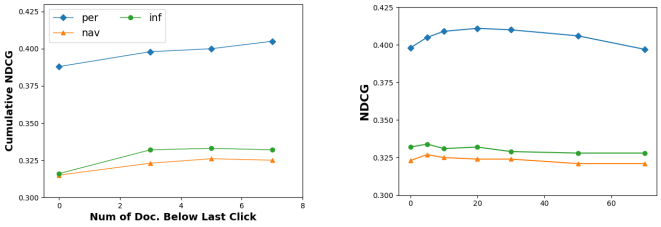


(a) Offline performance of linearly interpolating u_t and its projection g_t

(b) Cosine similarity between offline best model w^* and online model

(c) Comparing with ground-truth document space

Figure 2.7: Analyzing Document Space Projection.



(a) Including k documents following last clicked position

(b) Including r recently examined documents

Figure 2.8: Hyper-parameter tuning for Document Space Projection.

Analysis of Document Space Projection

To answer the second evaluation question, we design two experiments to show the effectiveness of document space projected gradient. In the first experiment, we study the utility of document space projected gradient. We compare the ranking performance of linearly interpolating the unrectified direction u_t and its document space projected version g_t, i.e., $\lambda g_t + (1-\lambda)u_t$, based on the MGD algorithm on MSLR-WEB10K dataset. Similar observations were obtained on other datasets, but due to space limit we have to omit those detailed results. We report the offline performance by varying λ from 0 (which is equivalent to the original MGD algorithm) and 1 (which is MGD-DSP) in Figure 2.7 (a). We can clearly observe a trend of increasing online performance over all three click models when we increase λ, i.e., trust more on the projected direction g_t for model update. This confirms the effectiveness of the projected direction g_t within document space comparing with the unrectified direction u_t from the entire parameter space. The offline performance is generally robust to the setting of λ for navigational and information click models. This is expected since both MGD and MGD-DSP are unbiased and will eventually converge to similar offline performance after sufficiently large number of iterations (we had 10,000 iterations in our experiments).

In the second experiment, we trained an offline LambdaRank model [57] using the complete annotated relevance labels in the large-scale MSLR-WEB10K dataset. Then given this w^*, we compared cosine similarity between the online estimated model parameters with and without DSP in each iteration using MGD as the baseline. We show the result of first 5,000 iterations. In Figure 2.7 (b) we can observe that MGD-DSP converges faster and better to w^* than MGD. This suggests the rectified gradient is more effective than the original one. We also compared with an oracle algorithm that knows the ground-truth examined documents, denoted as DSP-GT, to validate the effectiveness of our approximated document space. We show the result on DBGD and MGD under the perfect click model in Figure 2.7(c). We notice that oracle algorithms performed similarly as our proposed algorithm with an approximated document space, which confirms the effectiveness of the approximation heuristics.

To answer the third evaluation question, we compare different hyper-parameters used for constructing the document space on MSLR-WEB10K dataset. We vary k from 0 to 7 and report the result in Figure2.8 (a). We notice that for navigational and informational click models, a relatively small k achieved the best performance, i.e., $k = 3$. This corresponds to the observation that users do not continue to examine many documents after their last click under these two click models. However, under the perfect click model, the models' performance increases with a larger k. This aligns with the conclusions from our discussion in Section 2.2.3 that under the

perfect click model, we need to set a much larger k to accurately construct the document space and guarantee an unbiased gradient estimate.

In Figure 2.8(b), we vary r. As we discussed in Section 2.2.2, we are motivated to add recently examined documents to compensate for potentially overlooked examined documents in the current query. The effect of different choices of r is more noticeable under the perfect click model. This echoes our analysis above that under the perfect click model some examined documents may be overlooked when k is not large enough. Thus correctly setting up r could reduce the bias in document space construction and compensate the final performance. From the result figure, we notice that setting $r = 20$ provides the best result. Under navigational and informational click models, the algorithm is generally robust to the choice of r. This is because the approximations of examined documents are already accurate with a reasonable setting of k.

Chapter 3

Efficient Online Learning in Implicitly Structured Environments

We have introduced our efforts on leveraging explicit structural information to reduce sample complexity in Chapter 2. However, in practice such structural information may not be available to the learners. The challenge of learning by exploration in implicitly structured environments requires new algorithms that can infer necessary structural information during the online learning process. In this chapter, we present our research on efficient bandit learning algorithms in implicitly structured environments. In a low-rank collaborative environments, we develop factorization bandits that estimate latent factors on the fly. We also study that when the users are myopic, how the system can incentivize users' to explore without observing context features on the user side, i.e., under information gap. We develop an efficient incentive strategy such that the system can incentivize users to explore with such information disadvantage as long as the users' contexts has a linear transformation relation to the contexts on the system side.

3.1 Factorization Bandits for Implicit Collaborative Environment

Matrix factorization based collaborative filtering has become a standard practice in recommender systems [58–60]. The basic idea of such solutions is to characterize both recommendation items and users by vectors of latent factors inferred from *historical* user-item preference patterns via low-rank matrix completion [61, 62], with an assumption that only a few factors contribute to an individual's taste [58]. Despite a few recent advances in specific factorization techniques [63, 64], it is notoriously difficult to perform online interactive recommendation, because the need to focus on items that raise users' interest and, simultaneously, the need to explore new items for improving users' satisfaction in the long run create an explore-exploit dilemma. Periodically repeat model estimation to update latent factors is inept to handle the interactions between a system and its users on the fly, because not only does it overly exploit the learnt model that is biased towards previously frequently recommended items, but it also is prohibitively expensive to afford in terms of computational complexity.

Some preliminary attempts have been made to perform online matrix factorization for collaborative filtering. Basically, multi-armed bandit algorithms [9, 10] are employed to control the exploration of currently less promising recommendations for user feedback, and factorization is applied over the incrementally constructed user-item matrix on the fly. However, these two components are integrated in an *ad-hoc* manner: both contextual and context-free bandits have been explored on top of various factorization methods [29, 30, 65], given they only provide an index of candidate items for feedback acquisition. As a result, little is known about whether such combinations would lead to a converging recommendation performance nor would it ensure long-term optimality in theory, i.e., regret bound analysis.

We address the aforementioned challenges by performing online interactive recommendation by placing a factorization-based bandit algorithm on each user in the system. Low-rank matrix completion is performed

over an incrementally constructed user-item preference matrix, where an upper confidence bound (UCB) based item selection strategy is developed to balance the exploit/explore trade-off during online feedback acquisition. To better conquer cold-start in recommendation, two special treatments are devised. First, observable contextual features are integrated with the estimated latent factors during matrix factorization. This improves recommendation when the number of candidate items is large, but the payoffs are interrelated, i.e., *context-aware*. Second, the dependence among users (e.g., social influence) is introduced to our bandit algorithm through a collaborative reward generation assumption [1]. It enables information sharing among the neighboring users while online learning, so as to help reduce the overall regret.

More importantly, we rigorously prove that with high probability the developed algorithm achieves a sublinear upper regret bound for interactive recommendation, i.e., the average number of suboptimal recommendations made in our algorithm over time rapidly vanishes with high probability. And considerable regret reduction is achieved on both user and item sides because of our explicit modeling of observable contextual features and dependency among users. Extensive experimentations on both simulations and large-scale real-world datasets confirmed the advantages of the proposed algorithm compared with several state-of-the-art bandit-based factorization methods. Beyond recommender system, our factorization bandit solution is also extended to the social influence maximization task [5].

3.1.1 Related Work

There are some recent developments that focus on online collaborative filtering with multi-armed bandit algorithms, a reference solution for explore-exploit trade-off [9, 10, 12]. [30] studies interactive collaborative filtering via probabilistic matrix factorization. Both context-free and contextual bandit algorithms are introduced to perform online item selection based on the factorization results. [29] performs online low-rank matrix completion, where the explore/exploit balance is achieved via Thompson sampling. [65] introduces a UCB-like strategy to perform interactive collaborative filtering. The algorithm deterministically selects feedback user-item pairs using an index which depends on the covariance matrices of the posterior distributions of both latent user and item vectors. [66] performs co-clustering on users and items for collaborative filtering, where confidence bound on reward estimation is used to decide the clustering structures. However, because of the ad-hoc combinations of collaborative filtering methods and bandit methods in the aforementioned studies, limited theoretical understanding is available in those solutions. In this work, we provide a rigorous regret bound analysis of the developed factorization-based bandit algorithm, and demonstrate the algorithm's convergence property under different conditions. Moreover, our online factorization solution is general enough to incorporate several recent advances in factorization techniques, such as feature-based latent factor models [63, 64] and modeling mutual dependence among users [67, 68], which further improve the proposed algorithm's convergence rate during interactive online learning with users.

3.1.2 A Factorization Bandit Solution for Interactive Recommendation

Matrix factorization based collaborative filtering solutions map both users $U = \{u_1, u_2, ..., u_N\}$ and recommendation items $\mathcal{A} = \{a_1, a_2, ..., a_M\}$ to a joint latent factor space. The expected reward of an item with respect to a given user is assumed to be an inner product of the latent item factor $\mathbf{v}_a \in \mathbb{R}^l$ and the latent user factor $\boldsymbol{\theta}_u \in \mathbb{R}^l$. Hence, the reward generation process can be formalized as $r_{a,u} = \mathbf{v}_a^\intercal \boldsymbol{\theta}_u + \eta$, where the random variable η is drawn from a Gaussian distribution $N(0, \sigma^2)$. Regularized quadratic loss over a given set of user-item feedback pairs is usually employed to estimate the latent factors. Formally,

$$\min_{\boldsymbol{\theta}_u, \mathbf{v}_a} \frac{1}{2} \sum_{(a,u)\in\mathcal{K}} (\mathbf{v}_a^\intercal \boldsymbol{\theta}_u - r_{a,u})^2 + \frac{\lambda_1}{2} \sum_{u\in U} \|\boldsymbol{\theta}_u\|_2 + \frac{\lambda_2}{2} \sum_{a\in\mathcal{A}} \|\mathbf{v}_a\|_2 \tag{3.1}$$

where $\mathcal{K}$ is a set of user-item pairs with known reward (e.g., the offline training set), λ_1 and λ_2 are the trade-off parameters. The key research challenge in interactive matrix factorization is how to select the next feedback user-item pair for model update. Current practice exploits the trained model to collect user feedback, which unfortunately reinforces the bias in a currently inaccurate model. Therefore, properly explore some currently less promising items for model correction becomes necessary for long-term optimality.

Under the context of matrix factorization based collaborative filtering, the uncertainty of reward prediction comes from two sources: 1) the estimation error of latent user factors at trial t, i.e., $\|\hat{\boldsymbol{\theta}}_{u,t} - \boldsymbol{\theta}_u^*\|$, where $\hat{\boldsymbol{\theta}}_{u,t}$

is the current estimate of latent factors for user u, and $\boldsymbol{\theta}_u^*$ is the ground-truth factors; and 2) the estimation error of latent item factors at trial t, i.e., $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}_a^*\|$. Because of the regularized quadratic loss employed in Eq (3.1), the confidence sets of $\boldsymbol{\theta}_u$ and $\mathbf{v}_a$ estimation can be analytically computed [69], and thus readily be integrated to assemble a UCB-style bandit algorithm for interactive matrix factorization as follows,

$$\big(a_t, \langle \hat{\boldsymbol{\theta}}_{u,t}, \hat{\mathbf{v}}_{a,t} \rangle\big) = \underset{\big(a, \langle \boldsymbol{\theta}_u, \mathbf{v}_a \rangle\big) \in \mathcal{D}_t \times \mathcal{C}_{t-1}}{\arg\max} \boldsymbol{\theta}_u^\intercal \mathbf{v}_a \tag{3.2}$$

where $\mathcal{D}_t$ is the set of candidate items for recommendation at trial t, and $\mathcal{C}_{t-1}$ is the confidence set for latent user and item factors $\langle \boldsymbol{\theta}_u, \mathbf{v}_a \rangle$ constructed at last trial.

However, such a straightforward combination of bandit algorithm with matrix factorization cannot effectively solve the cold-start problem, as the estimation uncertainty of the latent factors for new users and new items is at the maximum. This inevitably requires more explorations on the new users and new items and hence leads to a decreased convergence rate of online learning and reduced user satisfaction in practice. We propose to solve these limitations by introducing observed contextual features [63, 70] and user dependence [1, 67] into online factorization. Both of these two techniques have been proved to be effective in offline matrix factorization, but little is known about their utility in an online setting. In particular, we explicitly incorporate these two components into our bandit algorithm's reward generation assumption, to make it a unified framework for interactive matrix factorization.

First, to reduce the reward prediction uncertainty on new items, we introduce observable contextual features into the estimation of latent item factors. Typical item-level contextual features include topic categories for news recommendation [12, 63] and genre for music recommendation [11]. Formally, we denote the observed contextual features for an item a as $\mathbf{x}_a \in \mathbb{R}^d$ and keep using $\mathbf{v}_a \in \mathbb{R}^l$ for its latent part (with $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 \leq L$). Accordingly, on the user side we redefine $\boldsymbol{\theta}_u = (\boldsymbol{\theta}_u^{\mathbf{x}}, \boldsymbol{\theta}_u^{\mathbf{v}}) \in \mathbb{R}^{d+l}$ (with $\|\boldsymbol{\theta}_u\|_2 \leq S$), in which $\boldsymbol{\theta}_u^{\mathbf{x}} \in \mathbb{R}^d$ corresponds to the context feature $\mathbf{x}_a$ and $\boldsymbol{\theta}_u^{\mathbf{v}} \in \mathbb{R}^l$ corresponds to the latent item factor $\mathbf{v}_a$. These extended user and item factors now determine the rewards in recommendation.

Second, we incorporate mutual influence among users to reduce the reward prediction uncertainty on new users. Distinct from existing solutions, where the dependency among users (such as social network) is introduced as graph-based regularization over the latent user factors [11, 67], we encode such dependency directly into our reward generation assumptions for matrix factorization. We assume the observed reward from each user is determined by a mixture of neighboring users [1]. Formally, instead of assuming N independent users for factorization, we place them on a weighted graph $G = (V, E)$, which encodes the affinity relation among users, to perform the estimation across them simultaneously. Each node V_u in G is parameterized by the latent user factor $\boldsymbol{\theta}_u$ for user u; and each edge in E represents the influence across users in reward generation. We encode this graph as an $N \times N$ stochastic matrix $\mathbf{W}$, in which each element w_{ij} is nonnegative and proportional to the influence that user j has on user i in determining the reward of different items. $\mathbf{W}$ is column-wise normalized such that $\sum_{j=1}^{N} w_{ij} = 1$ for $i \in \{1, ..., N\}$, and we assume $\mathbf{W}$ is time-invariant and known to the algorithm beforehand.

Based on the introduced contextual features and user relational graph G, we define a $(d + l) \times N$ matrix $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, which consists of latent user factors from all N users in graph G, and define $\mathbf{X}_{a_t} = (\mathbf{x}_{a_t,1}, ..., \mathbf{x}_{a_t,N})$ and $\mathbf{V}_{a_t} = (\mathbf{v}_{a_t,1}, ..., \mathbf{v}_{a_t,N})$ for the observable contextual features and latent item factors of the items to be presented to the N users respectively. To simplify the notations for discussion, we decompose $\boldsymbol{\Theta}$ into two sub-matrices, $\boldsymbol{\Theta}^{\mathbf{x}} = (\boldsymbol{\theta}_1^{\mathbf{x}}, \ldots, \boldsymbol{\theta}_N^{\mathbf{x}})$ and $\boldsymbol{\Theta}^{\mathbf{v}} = (\boldsymbol{\theta}_1^{\mathbf{v}}, \ldots, \boldsymbol{\theta}_N^{\mathbf{v}})$, corresponding to the observed context features and latent factors for items. As a result, we enhance our reward generation assumption as follows,

$$r_{a_t,u} = (\mathbf{x}_{a_t}, \mathbf{v}_{a_t})^\intercal \boldsymbol{\Theta} \mathbf{w}_u + \eta_t = \mathbf{x}_{a_t}^\intercal \boldsymbol{\Theta}^{\mathbf{x}} \mathbf{w}_u + \mathbf{v}_{a_t}^\intercal \boldsymbol{\Theta}^{\mathbf{v}} \mathbf{w}_u + \eta_t \tag{3.3}$$

Intuitively, in Eq (3.3) not only the observed contextual features, but also the estimated latent factors will be propagated through the user graph to determine the expected reward of items across users.

We will prove such information sharing greatly reduces sample complexity in learning the latent factors for both users and items. Plugging the enhanced reward generation assumption defined in Eq (3.3) into the regularized quadratic loss function in Eq (3.1), we can easily derive the closed-form solutions for $\boldsymbol{\Theta}$ and $\mathbf{v}_a$ *after* trial t via the alternating least square (ALS) method as $\vec{\boldsymbol{\Theta}}_t) = \mathbf{A}_t^{-1}\mathbf{b}_t$ and $\hat{\mathbf{v}}_{a,t} = \mathbf{C}_{a,t}^{-1}\mathbf{d}_{a,t}$, where the

detailed computation of $(\mathbf{A}_t, \mathbf{b}_t, \mathbf{C}_{a,t}, \mathbf{d}_{a,t})$ can be found in Algorithm 3. $\mathbf{I}_1$ and $\mathbf{I}_2$ are identity matrices with dimensions of $(d+l)N \times (d+l)N$ and $l \times l$ respectively. We define $\mathring{\mathbf{X}}_{a_t}$ as a special case of $\mathbf{X}_{a_t}$: only the column corresponding to user u is set to $\mathbf{x}_{a_t,u}$ and all the other columns are zero; and the same notation applies to $\mathring{\hat{\mathbf{V}}}_{a_t}$.

Under our enhanced reward generation assumption defined in Eq (3.3), the confidence set of $\langle \boldsymbol{\theta}_u, \mathbf{v}_a \rangle$ estimation can be analytically computed by the following lemma.

Lemma 3. *With proper initialization of ALS, the Hessian matrix of Eq* (3.1) *is positive definite at the optimizer* $\boldsymbol{\Theta}^*$ *and* $\mathbf{v}_a^*$*, such that for any* $\epsilon_1 > 0$*,* $\epsilon_2 > 0$*, and* $\delta \in (0,1)$*, with probability at least* $1-\delta$*, the estimation error of latent user and item factors satisfies,*

$$\|\vec{\hat{\boldsymbol{\Theta}}}_t) - \vec{\boldsymbol{\Theta}}^*)\|_{\mathbf{A}_t} \leq \sqrt{\log\big(\frac{det(\mathbf{A}_t)}{\delta\lambda_1}\big)} + \sqrt{\lambda_1}S + \frac{2}{\sqrt{\lambda_1}}\frac{(q_1+\epsilon_1)(1-(q_1+\epsilon_1)^t)}{1-(q_1+\epsilon_1)} \tag{3.4}$$

$$\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}_a^*\|_{\mathbf{C}_{a,t}} \leq \sqrt{\log\big(\frac{det(\mathbf{C}_{a,t})}{\delta\lambda_2}\big)} + \sqrt{\lambda_2}L + \frac{2}{\sqrt{\lambda_2}}\frac{(q_2+\epsilon_2)(1-(q_2+\epsilon_2)^t)}{1-(q_2+\epsilon_2)} \tag{3.5}$$

in which $q_1 \in (0,1)$ *and* $q_2 \in (0,1)$.

In Lemma 3, ϵ_1 and ϵ_2 are the precision parameters for ALS, and q_1 and q_2 can be explicitly estimated as described in [71]. The key assumption behind this lemma is the noise distribution in reward generation defined in Eq (3.3) is *stationary*. As a result, this lemma gives us a reasonable construction of the confidence sets for $\boldsymbol{\Theta}$ and $\mathbf{v}_a$ estimation, which can be easily transformed to the estimation uncertainty of payoff $r_{a_t,u}$. The proof sketch of this lemma can be found in the appendix.

Based on Lemma 3, we define α_t^u and α_t^a as the upper bound of $\|\vec{\hat{\boldsymbol{\Theta}}}_t) - \vec{\boldsymbol{\Theta}}^*)\|_{\mathbf{A}_t}$ and $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}_a^*\|_{\mathbf{C}_{a,t}}$ respectively. By applying the UCB principle, the item selection strategy for our bandit algorithm can be derived as step 9 in Algorithm 3. In particular, the first term in our item selection strategy is an online prediction of the expected reward based on the current estimation of latent user factors and item factors. It reflects the tendency for exploiting the current estimates. The second and third terms are related to the estimation uncertainty of $\mathbf{v}_a$ and $\boldsymbol{\Theta}$. They reflect the tendency for exploring currently less promising but highly uncertain items. It is easy to verify that the exploration terms shrink when more observations become available, such that the exploit/explore trade-off is balanced dynamically. Later on we prove that because of the explicit modeling of user dependency (i.e., Eq (3.3)), the exploration term also uniformly shrinks for new users and new items, which lead to considerable regret reduction over all users. We name the resulting bandit algorithm as FactorUCB, and illustrate the detailed procedure of it in Algorithm 3.

3.1.3 Regret Analysis

To quantify the performance of factorUCB, we consider the accumulated (pseudo) regret defined Eq (2.1). Based on Lemma 3 and the developed item selection strategy, we have the following theorem about the upper regret bound of FactorUCB algorithm.

Theorem 4. *Under proper initialization of ALS in Algorithm 3, with probability at least* $1-\delta$*, the accumulated regret of FactorUCB algorithm satisfies,*

$$\begin{aligned}\boldsymbol{R}(T) \leq & 2\alpha_T^u\sqrt{2(d+l)NT\log\Big(1+\frac{L^2\sum_{t=1}^T\sum_j^N w_{u_t,j}^2}{\delta\lambda_1(d+l)N}\Big)} + 2\alpha_T^a\sqrt{2lT\log\Big(1+\frac{S^2\sum_{t=1}^T\sum_j^N w_{u_t,j}^2}{\delta\lambda_2 l}\Big)} \\ & + 2\alpha_T^a\frac{(q_2+\epsilon_2)\big(1-(q_2+\epsilon_2)^T\big)}{1-(q_2+\epsilon_2)}\end{aligned} \tag{3.6}$$

in which q_2 and ϵ_2 are the same as those defined in Lemma 3, α_T^u and α_T^a are the upper bound of $\|\vec{\hat{\boldsymbol{\Theta}}}_t) - \vec{\boldsymbol{\Theta}}^*)\|_{\mathbf{A}_t}$ and $\|\hat{\mathbf{v}}_{a,t} - \mathbf{v}_a^*\|_{\mathbf{C}_{a,t}}$ over all $t \in \{1,\ldots,T\}$ respectively, and δ is also encoded in α_T^u and α_T^a as shown in Eq (3.4) and (3.5). Though required by the theorem that λ_1 and λ_2 have to be sufficiently large, in our empirical evaluations the algorithm's performance is not sensitive to this setting. The specific form of α_T^u and α_T^a and the proof sketch of this theorem are provided in the appendix.

Algorithm 3 FactorUCB

Inputs: $\lambda_1, \lambda_2 \in (0, +\infty), l \in \mathbb{Z}^+$
Initialize: $\mathbf{A}_1 \leftarrow \lambda_1 \mathbf{I}_1$, $\mathbf{b}_1 \leftarrow \mathbf{0}^{(d+l)N}$, $\vec{\mathbf{\Theta}}_1) \leftarrow \mathbf{A}_1^{-1}\mathbf{b}_1$
for $t = 1$ to T **do**
  Receive user u_t
  Observe feature vectors, $\mathbf{x}_a \in \mathbb{R}^d$
  if item a is new **then**
    initialize $\mathbf{C}_{a,t} \leftarrow \lambda_2 \mathbf{I}_2$, $\mathbf{d}_{a,t} \leftarrow \mathbf{0}^l$, $\hat{\mathbf{v}}_{a,t} \leftarrow \mathbf{0}^l$
  Select item by $a_t = \arg\max_{a_{\in}\mathcal{A}} \Big((\mathbf{x}_a, \hat{\mathbf{v}}_{a,t})^\intercal \hat{\mathbf{\Theta}}_t \mathbf{w}_{u_t} + \alpha_t^u \sqrt{vec\big((\mathring{\mathbf{X}}_{a_t}, \mathring{\hat{\mathbf{V}}}_{a_t})\mathbf{W}^\intercal\big)\mathbf{A}_t^{-1} vec\big((\mathring{\mathbf{X}}_{a_t}, \mathring{\hat{\mathbf{V}}}_{a_t})\mathbf{W}^\intercal\big)^\intercal} \Big) + \alpha_t^a \sqrt{(\hat{\mathbf{\Theta}}_t \mathbf{w}_{u_t})\mathbf{C}_{a,t}^{-1}(\hat{\mathbf{\Theta}}_t \mathbf{w}_{u_t})^\intercal}$
  Observe reward r_{a_t,u_t} from user u_t
  $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + (\vec{\mathring{\mathbf{X}}}_{a_t}, \mathring{\hat{\mathbf{V}}}_{a_t})\mathbf{W}^\intercal)(\vec{\mathring{\mathbf{X}}}_{a_t}, \mathring{\hat{\mathbf{V}}}_{a_t})\mathbf{W}^\intercal)^\intercal$
  $\mathbf{b}_{t+1} \leftarrow \mathbf{b}_t + (\vec{\mathring{\mathbf{X}}}_{a_t}, \mathring{\hat{\mathbf{V}}}_{a_t})\mathbf{W}^\intercal) r_{a_t,u_t}$
  $\vec{\mathbf{\Theta}}_{t+1}) \leftarrow \mathbf{A}_{t+1}^{-1}\mathbf{b}_{t+1}$
  $\mathbf{C}_{a_t,t+1} \leftarrow \mathbf{C}_{a_t,t} + (\hat{\mathbf{\Theta}}_t^{\mathbf{y}} \mathbf{w}_{u_t})(\hat{\mathbf{\Theta}}_t^{\mathbf{y}} \mathbf{w}_{u_t})^\intercal$
  $\mathbf{d}_{a_t,t+1} \leftarrow \mathbf{d}_{a_t,t} + (\hat{\mathbf{\Theta}}_t^{\mathbf{y}} \mathbf{w}_{u_t})(r_{a_t,u_t} - \mathbf{x}_{a_t}^\intercal (\hat{\mathbf{\Theta}}_t^{\mathbf{x}} \mathbf{w}_{u_t}))$
  $\hat{\mathbf{v}}_{a_t,t+1} \leftarrow \mathbf{C}_{a_t,t+1}^{-1} \mathbf{d}_{a_t,t+1}$
  Project $\hat{\mathbf{\Theta}}_{t+1}$ and $\hat{\mathbf{v}}_{a_t,t+1}$ with respect to the constraints $\|\boldsymbol{\theta}_u\|_2 \leq S$ and $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 \leq L$

As highlighted in the proof, because the confidence interval is shrinking via exploration, a sublinear regret is achieved after T trials of interactions; otherwise without proper exploration, such as in the conventional offline training and online testing paradigm of matrix factorization, a linear regret is inevitable. Moreover, the resulting regret bound of factorUCB has the following important theoretical properties under different conditions.

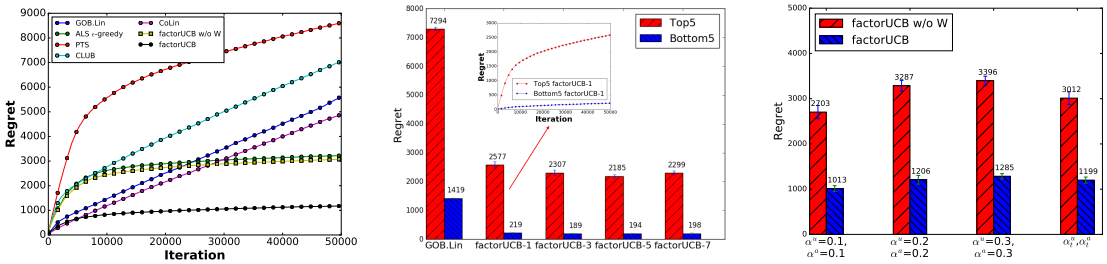
First, the dependency structure among users plays an important role in reducing the regret on both user side and item side. Consider the following two extreme cases. In the first case, when $\mathbf{W}$ is an identity matrix, i.e., no dependency among users, the first term of the upper regret bound in Eq (3.6) degenerates to $O\big(N(d+l)\sqrt{T}\log\frac{T}{N}\big)$, which roots in the reward prediction uncertainty from the estimated latent user factors. And the second term degenerates to $O\big(l\sqrt{T}\log T\big)$, which corresponds to the estimated latent item factors. In the second case, when users are homogenous and have uniform influence among each other, i.e., $\forall i, j, w_{ij} = \frac{1}{N}$, the first term in the regret bound decreases to $O\big(N(d+l)\sqrt{T}\log\frac{T}{N^2}\big)$ and the second decreases to $O\big(l\sqrt{T}\log\frac{T}{N}\big)$. As a result, via modeling user dependency, FactorUCB achieves an $O\big(N(d+l)\sqrt{T}\log N\big)$ regret reduction on the user side and an $O\big(l\sqrt{T}\log N\big)$ regret reduction on the item side.

Second, as denoted in Eq (3.6), the user arrival sequence is recorded in the summation term of $\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t,j}^2$, which is bounded by T from above, no matter how users arrive to the system (as $\mathbf{w}_u$ is a stochastic vector). Therefore, the upper regret bound of factorUCB stays in $O\big(N(d+l)\sqrt{T}\log\frac{T}{N}\big)$ in the worse case scenario, such as users arrive in an adversarial way – the least connected users come first and most often.

Third, following our enhanced reward generation assumption specified in Eq (3.3), the estimation quality of latent user factors in factorUCB satisfies the following inequality (similar result applies to the estimation quality of latent item factors as well),

$$\|\vec{\mathbf{\Theta}}_t) - \vec{\mathbf{\Theta}}^*)\|_{\mathbf{A}_t} \leq \sqrt{\log\big(\frac{det(\mathbf{A}_t)}{\delta\lambda_1}\big)} + \sqrt{\lambda_1}S + \frac{2}{\sqrt{\delta\lambda_1}}\sum_{t'=1}^{t}\|\mathbf{v}_{a_{t'},u}^* - \hat{\mathbf{v}}_{a_{t'},u}\|_2 \tag{3.7}$$

If the dimension of latent factors matches the ground-truth, based on the proved convergence property of ALS in [71], the estimation of $\mathbf{\Theta}$ and $\mathbf{v}_a$ is q-linearly convergent to the optimum $(\mathbf{\Theta}^*, \mathbf{v}_a^*)$, which is the conclusion in Lemma 3. But if the dimension is not correctly set and those latent factors are independent from each other, the third term in Eq (3.7) will not converge. It makes α_t^u linearly increase over time as α_t^u is the upper bound of $\|\vec{\mathbf{\Theta}}_t) - \vec{\mathbf{\Theta}}^*)\|_{\mathbf{A}_t}$. This leads to a linear regret in factorUCB at the worst case. Admittedly, determining the dimension of latent factors is always a bottleneck of factorization-based methods in practice.



(a) Cumulated regret (b) Effect of latent dimensions (c) Effect of (α_t^u, α_t^a)

Figure 3.1: Analysis of regret, hidden feature dimension and parameter tuning.

But by introducing the observable contextual features, especially those strongly correlated with the expected rewards, the reward prediction uncertainty can be reduced as the latent factors only need to fit the residual of reward prediction from the observed features (as shown in the estimation of $\mathbf{v}_a$ in Algorithm 3). This leads to reasonable performance of factorUCB in our empirical evaluations.

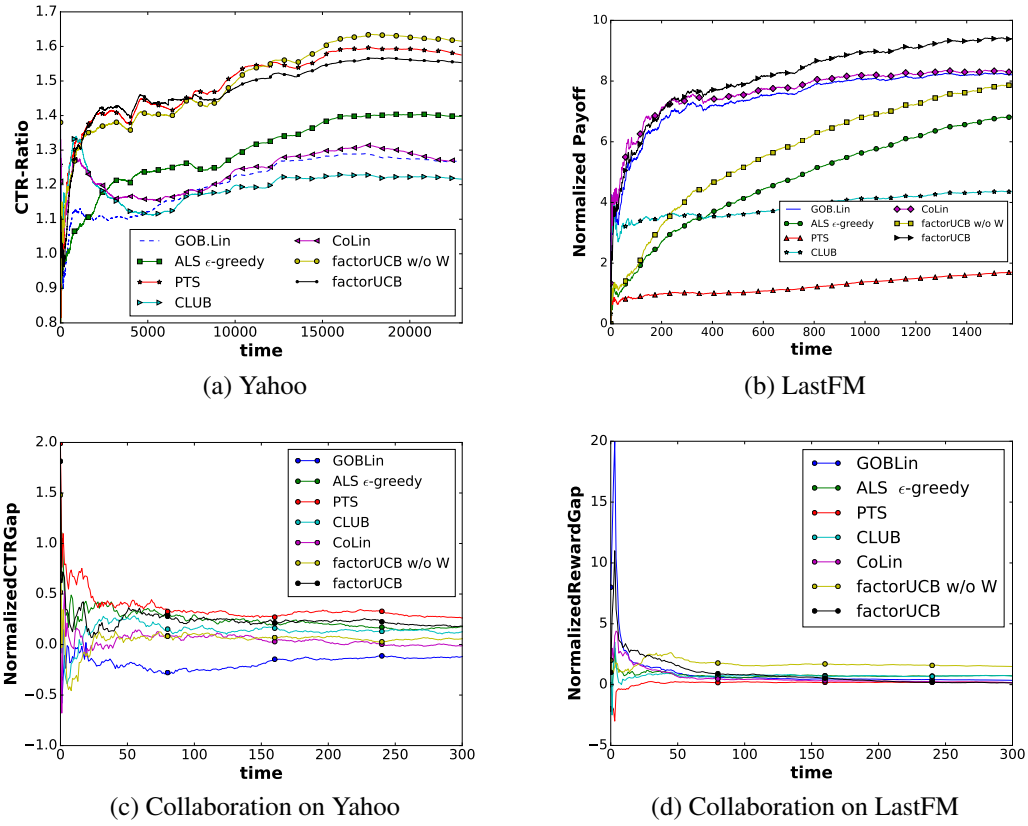
3.1.4 Experiments

We performed extensive empirical evaluations of our proposed factorUCB algorithm against several state-of-the-art factorization-based and bandit-based collaborative filtering methods, including: 1) Alternating Least Square (ALS) with ϵ-greedy [30], which applies context-free ϵ-greedy algorithm based on both observed features and latent factors, but cannot utilize the user relational graph; 2) Particle Thompson Sampling for matrix factorization (PTS) [29], which combines Thompson sampling with probabilistic matrix factorization based on Rao-Blackwellized particle filter, and it cannot utilize observed features and user relational graph; 3) GOB.Lin [11], which models the dependency among a set of contextual bandits over users via a graph Laplacian based model regularization, but cannot estimate the latent factors; 4) CLUB [25], which clusters users during online learning to enable model sharing; but it only works with contextual features; 5) CoLin [1], which imposes a similar collaborative reward generation assumption over the user relational graph as that in our algorithm, but does not model latent factors; 6) factorUCB w/o W, which is factorUCB with an identity $\mathbf{W}$ matrix, i.e., the dependency among users is not considered; it demonstrates of utility of modeling user dependency in interactive recommendation.

Experiments on synthetic dataset

In simulation, we generated a size-K item pool $\mathcal{A}$, in which each item a is associated with a $(d+l)$-dimension feature vector $(\mathbf{x}_a, \mathbf{v}_a)$. Each dimension is drawn from a set of zero-mean Gaussian distributions with variances sampled from a uniform distribution $U(0,1)$. Principle Component Analysis (PCA) was performed to make all the dimensions *orthogonal* to each other. To simulate the reward generation defined in Eq (3.3), we used all the $(d+l)$-dimension features to compute the true reward for each item, but only revealed the first d dimensions (i.e., $\mathbf{x}_a$) to an algorithm. We simulated N users, each of who is associated with a $(d+l)$-dimension preference vector $\boldsymbol{\theta}_u^*$. Each dimension of $\boldsymbol{\theta}_u^*$ is drawn from a uniform distribution $U(0,1)$. $\boldsymbol{\theta}_u^*$ is treated as the ground-truth latent user factor in reward generation, and is unknown to the algorithms. We then constructed the golden relational stochastic matrix $\mathbf{W}$ for the dependency graph of users by defining $w_{ij} \propto \langle \boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^* \rangle$, and normalize each column of $\mathbf{W}$ by its L1 norm. The resulting $\mathbf{W}$ was disclosed to all the algorithms. To increase the learning complexity, at each trial t, our simulator only disclosed a subset of items in $\mathcal{A}$ to the learners for selection, e.g., randomly selected 10 items from $\mathcal{A}$ without replacement. At each trial t, the same set of items were presented to all the algorithms; and the Gaussian noise η_t in Eq (3.3) was sampled once for all those items at each trial. We fixed the dimension d of observable features to 20, the dimension l of latent item factors to 5, user size N to 100, the standard derivation σ of Gaussian noise to 0.1, and the item pool size K to 1000 in our simulation.

Cumulated regret defined in Eq (2.1) was used to evaluate the performance of different algorithms in Figure 3.1 (a), where we set the dimension for latent factors in PTS to 10 (which gave us the best performance) and 5 in ALS ϵ-greedy and factorUCB. We observed that PTS took much longer time to converge, because PTS cannot utilize the observed context features for reward prediction, so that it requires much more observations to improve the accuracy of latent factor estimation. Instead, ALS ϵ-greedy and factorUCB leveraged the context features to quickly reduce the reward prediction uncertainty (i.e., less exploration). Two contextual bandits, i.e.,



(a) Yahoo (b) LastFM

(c) Collaboration on Yahoo (d) Collaboration on LastFM

Figure 3.2: Experimental comparisons on real-world datasets.

GOB.Lin and CoLin, suffered from linear regret, since they do not model the latent item factors. In addition, factorUCB converged much faster than factorUCB w/o W, which confirmed our theoretical analysis about the regret reduction from user dependency modeling.

Because factorUCB requires the dimension of latent factor as input, we test its sensitivity to the setting of latent dimension l. To investigate the importance of correct setup of latent factor dimension in factorUCB, we tested two different ways of latent factor construction in our simulator: 1) we chose the top 5 dimensions with the largest eigenvalue from PCA's result as latent item factors, i.e., we hid the top 5 most informative factors in reward generation from the learners; 2) we hid the bottom 5 most informative factors. And on the algorithm side, we varied the dimension of latent factors used in factorUCB from 1 to 7. From the results shown in Figure 3.1 (b), we can reach three conclusions. First, when the latent factors were the most informative ones, we obtained much worse regret than that in the case of the least informative factors were hidden. Second, the large difference between the regret of a bandit algorithm that does not model the latent factors (such as GOB.Lin) and the one that models latent factors (factorUCB, even with wrong dimensions) emphasizes the necessity of latent factor learning in online recommendation. Third, although our theoretical analysis predicts a linear regret if the latent factor dimension was not accurately set, the actual performance was much more promising. One reason is that our theoretical analysis is for the worst case scenario (upper regret bound), which does not preclude a sub-linear converging regret in practice.

In addition, we also investigated the effect of exploration parameter α_t^u and α_t^a in factorUCB, compared with factorUCB w/o W. In Figure 3.1 (c), each column illustrates a combination of α_t^u and α_t^a used in factorUCB and factorUCB w/o W. The last column indexed by (α_t^u, α_t^a) represents the theoretical values of those two parameters computed from the algorithm's corresponding regret analysis. As shown in the results, the empirically tuned (α^u, α^a) yielded comparable performance to the theoretical values, and made online computation more efficient. As a result, in all our following experiments we will use the manually set α_t^u and α_t^a.

Experiments on real-world datasets

Yahoo dataset: We reported the normalized CTR results from different algorithms over 160 derived user groups in Figure 3.2 (a) (similar relative improvement was obtained with different number of derived user groups). Both variants of FactorUCB outperformed conventional bandit algorithm (i.e., GOB.Lin, CoLin and CLUB) and factorization method (i.e., ALS ϵ-greedy). And clearly via modeling user dependency during online factorization, FactorUCB improves more rapidly than PTS when more observations become available.

LastFM dataset: We normalized the accumulated reward from different algorithms by that from a random algorithm, and reported the results in Figure 3.2 (b). We can clearly notice that PTS performed the worst, while two contextual bandits (i.e., GOB.Lin and CoLin) achieved much better performance than it. This indicates the observed context features in this dataset were sufficiently informative for the algorithms to make accurate recommendations. A purely factorization-based method got penalized by not utilizing such information. On the other hand, we also noticed that factorUCB converged much faster than factorUCB w/o W, which again demonstrates the utility of user dependency modeling for addressing cold-start in recommendation.

To further investigate the effect of modeling context features and user dependency in alleviating cold-start in recommendation, we designed a set of controlled experiments. We first split users into two groups using a max-cut algorithm on the constructed user relational graph to maximize the connectivity between these two groups. Observations in the first user group are called "learning group" and those in the second group are called "testing group." To simulate cold-start, we only executed algorithms on the testing group. Correspondingly, we simulated *warm-start* by first running algorithms on the learning group to pre-estimate the models, and then continuing them on the testing group. Since users in the testing group were isolated from the learning group, their model parameters could only be initialized by the propagated information via the user relational graph, if an algorithm explicitly modeled that.

We measured the differences in average CTR on Yahoo and accumulated rewards on LastFM between *warm-start* and *cold-start* in Figure 3.2 (c) and (d). On the Yahoo dataset, factorization-based algorithms (i.e., factorUCB, PTS and ALS ϵ-greedy) benefit the most from the collaboration in latent factor estimation: latent item factors estimated in the learning group helped them better estimate user preferences in testing group. On the LastFM dataset, considerable improvement was achieved in algorithms explicitly modeling user dependency, i.e., factorUCB, GOB.Lin and CoLin.

3.2 Incentivize Exploration Under Information Gap

Classical bandit research studies the single-party setting, where the system has a full control over which arm to pull. However, in many real-world applications, such as recommender systems and e-commerce platforms, one often faces a *two-party* game between the system and its users, who have *different* interests and roles in this game. Specifically, the system aims at maximizing long-term cumulative reward, which requires exploration in the arm space. However, the decision about which arm to pull is made by the users, and the system can only observe the rewards associated with the users' decisions. The users often act as *myopic* agents, who only seek to maximize their short-term utilities, i.e., exploit the arm with the currently best estimated reward. This division leads to the problems of under-exploration and selection bias: the best arm may remain unexplored forever if it appears sub-optimal initially. To align the two parties' interest, the system has to offer compensations to users so that they are motivated to try the exploratory arms, which in turn helps system maximize long-term cumulative reward. This problem is known as *incentivizied exploration* [72–74].

The system's goal in incentivized exploration is to minimize total compensation while maximizing cumulative rewards [73, 75, 76]. Existing solutions assume both parties maintain the same reward estimation. This assumption is necessary for the system to compute the compensation based on the users' estimated reward difference between the currently best arm and the exploratory arm. Under a context-free setting (aka Multi-armed Bandit (MAB) [7, 8] in literature), this assumption naturally holds because both parties maintain the same estimated mean reward on each pulled arm. And most existing incentivized exploration solutions work under this setting. However, under the contextual bandit setting [10, 12, 69], the two parties may associate the same observed rewards with *different context features*. For example, in a recommender system, the users could access features related to their own private information (e.g., gender and age), which are not accessible by the system. To obtain the same quality of reward estimation, the system has to resort to other behavior features

that profile such private user features [77, 78]. This situation can be easily understood by an extreme case in a finite arm setting: the system only observes the index of each arm, while the users employ informative features of the arms. As a result, the system suffers from a much slower convergence rate in reward estimation than the users. We refer to this representation asymmetry as the *information gap* between the two parties, which brings in new challenges to incentivized exploration. For example, the system no longer knows which arm has the best estimated reward on the user side.

In this work, we propose an algorithm that incentivizes the users to explore according to the Linear UCB strategy [12, 69] under the information gap. Our key idea is that although the system suffers from an information disadvantage and cannot compute the minimum compensation precisely, offering a larger amount of compensation guarantees sufficiency for users to explore. And this added compensation should shrink fast enough such that the total compensation is still sublinear. We prove that in T rounds of interaction our algorithm achieves compensation and regret both in the order of $O(d_v\sqrt{T}\log T)$ with information gap and $O(d_x\sqrt{T}\log T)$ without information gap, where d_x and d_v are the dimensions of context features used by the users and the system, respectively. The results suggest that incentivized exploration is still possible with information gap, and the added cost is realized by the extra compensation that is dominated by d_v. We also prove the compensation lower bound of incentivized exploration in linear contextual bandits, which generalizes the result of compensation lower bound in MAB settings reported in [76]. Our simulation-based empirical studies also validate the effectiveness and cost-efficiency of the proposed algorithm.

3.2.1 Related Work

The incentivized exploration problem in multi-armed bandits has been studied since [72, 73]. See [79] for an overview. One line of the studies assume the system has information advantage on observing the full arm-pulling history while users do not [72, 74, 80, 81]. The system leverages the information asymmetry to recommend exploratory arms as long as the users do not have a better choice from their perspective. Another line considered the setting where the arm-pulling history is publicly available to both system and users and the system offers compensations to an arm for incentivized exploration [73, 76, 82]. Our setting follows this line of research.

Incentivized learning with monetary payments was first studied in [73] in a Bayesian setting with discounted regret and compensation. Chen et al. [82] studied a heterogeneous users setting, where user diversity led to their solution with constant compensation. Agrawal et al. [83] considered heterogeneous contexts in a contextual bandit setting. In [76], the authors analyzed the non-Bayesian and non-discounted reward case and showed $O(\log T)$ regret and compensation in a stochastic MAB setting. Liu et al. [84] considered the reward feedback is biased because of the compensation. Kannan et al. [85] considered incentivized exploration for fair recommendation. Our setting is mostly similar to [76], i.e., non-Bayesian and non-discounted reward, but is studied under the linear contextual bandit setting. We should note all the aforementioned studies assume the system and the users share the same information such as arm pulls, rewards and contexts, and the system calculates the compensation based on the shared information. Our setting is strictly more challenging. The information gap is caused by information asymmetry: the system cannot access the feature vectors employed by the users. As a result, users' reward estimation will be different from the system' and the precise amount of payment is harder to compute.

There are several recent works study low-rank bandits, which however are intrinsically different from ours. For example, Lale et al. [86] consider the contexts are sampled from a low-dimensional subspace and propose a PCA-based solution to reduce the dimensionality. Yang et al. [87] study multi-task linear bandits with a shared low-rank structure. These methods assume the learning problem is generated from a low-rank structure but presented in a high-dimensional space. But in our setting, the system's observed contexts are already sampled from a high-dimensional compact space, whose dimension cannot be further reduced. The information gap in representation asymmetry is a unique problem in this two-party game setting.

3.2.2 Problem Definition

Notations and assumptions. We study the problem under a linear bandit setting, where a myopic user sequentially interacts with the system for T rounds. At each round t, the user observes compensation offered

by the system, and pulls an arm a_t from a given arm set $\mathcal{A}_t$. Both the system and the user observe the resulting reward $r_{a_t,t}$ and update their estimations accordingly.

In a contextual bandit setting, each arm a is associated with a context feature vector. In our problem, for arm $a \in \mathcal{A}_t$, the system observes a feature vector $\mathbf{v}_a$ from a d_v-dimensional subspace and the users observe a feature vector $\mathbf{x}_a$ from a d_x-dimensional subspace. Without loss of generality, we will assume $\mathbf{x}_a \in \mathbb{R}^{d_x}$ and $\mathbf{v}_a \in \mathbb{R}^{d_v}$ — if not, the standard PCA technique can be used to reduce the feature dimensions to d_x and d_v [86]. Essentially we consider the features span the whole vector space respectively, which means there is no feature without support on both sides and the dimensionality cannot be further reduced.

Assumption 1 (Information Gap)**.** *There exists a linear transformation $P \in \mathbb{R}^{d_x \times d_v}$ (where $d_v \geq d_x$) such that for any arm a,*

$$\mathbf{x}_a = P\mathbf{v}_a \tag{3.8}$$

The assumption on $d_v \geq d_x$, i.e., features used on the user side belong to a lower dimension space, is motivated by many real-world scenarios: for example, users can construct features related to their private information (e.g., age, gender, income or health), while the system has to employ a lot of behavioral features to resemble such information [77, 78]. To better illustrate the concept, we describe a few real-world examples below where the gap exists and is inevitable.

Examples of information gap. A notable special case of linear bandits with information gap is a K-armed contextual bandit problem, where the system knows nothing beyond the indices of arms. In this case, the context vectors used by the system are the K-dimension one-hot vectors, while the user may employ d_x-dimension feature representations of the same arms. The information gap ($K > d_x$) is encoded in the transformation matrix P. Now let us consider a less extreme example. Some features could be the combinations of both the user's private information and item's property, e.g., joint of user's income and the item's price, or joint of user's gender and the item's category. This is a typical way to construct features in practical recommender systems [12]. The users can employ these informative features and enjoy a faster reward estimation convergence; but the system suffers when it cannot access users' private information. In this example, the transformation matrix P contains the private information hidden from the system.

Note that having more features is not equivalent to having a more informative representation. Another practical example is that the context vectors used by the system may include many useless or redundant features, such that the corresponding weights in the ground-truth model parameter $\boldsymbol{\theta}_v^*$ are zeros, i.e., a sparse regression setting. In this example, the system's features are clearly less informative, because of the useless features.

The information gap between the two parties is characterized by matrix P. The linear transformation assumption is to guarantee the two parties face a linear reward mapping, which we state below.

Reward mapping. Following a linear bandit setting, the expected reward of arm a is determined by the inner product between the context features and unknown bandit model parameter. From the user's perspective, we have $\mathbb{E}[r_a] = \mathbf{x}_a^\mathsf{T}\boldsymbol{\theta}_x^*$ where $\boldsymbol{\theta}_x^*$ is the unknown model parameter on the user side. From Assumption 1, we have $\mathbf{x}_a^\mathsf{T}\boldsymbol{\theta}_x^* = \mathbf{v}_a^\mathsf{T}P^\mathsf{T}\boldsymbol{\theta}_x^*$, which suggests there always exists a parameter $\boldsymbol{\theta}_v^* = P^\mathsf{T}\boldsymbol{\theta}_x^*$ on the system side satisfying the same linear reward mapping. We summarize the reward mapping on the two sides as follow:

$$\mathbb{E}[r_a] = \mathbf{x}_a^\mathsf{T}\boldsymbol{\theta}_x^* = \mathbf{v}_a^\mathsf{T}\boldsymbol{\theta}_v^*$$

After the user pulls arm a_t, both sides observe the reward $r_{a_t,t}$ as

$$r_{a_t,t} = \mathbb{E}[r_{a_t}] + \eta_t \tag{3.9}$$

where η_t is R-sub-Gaussian noise. Without loss of generality, we assume that the norm of the features and parameters are bounded as $\|\mathbf{x}_a\|_2 \leq \|\mathbf{v}_a\|_2 \leq 1$, $\|\boldsymbol{\theta}_x^*\|_2 \leq 1$, $\|\boldsymbol{\theta}_v^*\|_2 \leq 1$, which naturally bounds the expected reward in the range of $[-1, 1]$ and simplifies the analysis. Note that the assumption of $\|\mathbf{x}_a\|_2 \leq \|\mathbf{v}_a\|_2$ is equivalent as assuming the largest singular value of P is upper bounded by 1. Intuitively, this means the linear

Algorithm 4 Incentivized LinUCB under Information Gap

Inputs: λ, δ
Initialize: $\mathbf{A}_x = \lambda \mathbf{I}_{d_x}, \mathbf{A}_v = \lambda \mathbf{I}_{d_v}, \mathbf{b}_x = 0, \mathbf{b}_v = 0$
for $t = 1$ to T **do**
  System and user observe context vectors $\{\mathbf{v}_a\}_{a \in \mathcal{A}_t}$ and $\{\mathbf{x}_a\}_{a \in \mathcal{A}_t}$ respectively
  System calculates compensation $c_{a,t}$ for arm a according to Eq (3.13)
  User pulls arm $a_t = \arg\max_{a \in \mathcal{A}} \hat{r}_{x,a,t} + c_{a,t}$
  System and user observe reward r_{a_t}
  // Update on the system side:
  $\mathbf{A}_{v,t+1} \leftarrow \mathbf{A}_{v,t} + \mathbf{v}_{a_t} \mathbf{v}_{a_t}^\mathsf{T}$, $\mathbf{b}_{v,t+1} \leftarrow \mathbf{b}_{v,t} + \mathbf{v}_{a_t} r_{a_t}$
  $\hat{\boldsymbol{\theta}}_{v,t+1} \leftarrow {\mathbf{A}_{v,t+1}}^{-1} \mathbf{b}_{v,t+1}$
  // Update on the user side:
  $\mathbf{A}_{x,t+1} \leftarrow \mathbf{A}_{x,t} + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^\mathsf{T}$, $\mathbf{b}_{x,t+1} \leftarrow \mathbf{b}_{x,t} + \mathbf{x}_{a_t} r_{a_t}$
  $\hat{\boldsymbol{\theta}}_{x,t+1} \leftarrow \mathbf{A}_{x,t+1}^{-1} \mathbf{b}_{x,t+1}$

transformation does not amplify the magnitude of the features. One can always find the satisfying $\mathbf{x}_a$ by re-scaling $\boldsymbol{\theta}_x^*$ accordingly.

The system and the user estimate their own model parameters using ridge regression separately, denoted as $\hat{\boldsymbol{\theta}}_{v,t}$ and $\hat{\boldsymbol{\theta}}_{x,t}$, by the same observed rewards $\{r_{a_t,t}\}$ but different context features. As a result, the two parties would predict different rewards for the same arm a, denoted as $\hat{r}_{x,a,t} = \mathbf{x}_a^\mathsf{T} \hat{\boldsymbol{\theta}}_{x,t}$ and $\hat{r}_{v,a,t} = \mathbf{v}_a^\mathsf{T} \hat{\boldsymbol{\theta}}_{v,t}$.

Objective. The users and the system have different objectives in this sequential decision making problem: the user aims to maximize his/her short-term instantaneous reward, while the system aims to maximize the long-term cumulative reward. At each round t, without any incentive, a myopic user will exploit the arm with the highest estimated reward, i.e., $a = \arg\max_{i \in \mathcal{A}_t} \hat{r}_{x,i,t}$. It is well known that such exploitation-only decisions will lead to sub-optimal cumulative reward in the long term. In order to balance exploitation and exploration, the system has to provide compensations to encourage the user to explore. Specifically, the system offers compensation $c_{a,t}$ for pulling arm a. Given the incentives, the user maximizes the instantaneous utility by pulling arm $a_t = \arg\max_{i \in \mathcal{A}_t} \hat{r}_{x,i,t} + c_{i,t}$.

The system seeks to maximize the cumulative reward, or equivalently, minimize the *cumulative regret* while also minimizing the *total compensation* in expectation. The system's regret is defined as

$$R(T) = \sum_{t=1}^{T} \big(\mathbb{E}[r_{a_t^*}] - \mathbb{E}[r_{a_t}] \big) \tag{3.10}$$

where a_t^* is the optimal arm with the highest expected reward at time t. The total compensation is defined as

$$C(T) = \sum_{t=1}^{T} \mathbb{E}[c_{a_t,t}] \tag{3.11}$$

An effective incentivized exploration method should balance the trade-off among exploration, exploitation and compensation to obtain *sublinear* cumulative regret and *sublinear* total compensation.

3.2.3 Incentivized Exploration in Linear Bandits

We present our solution on incentivized exploration under information gap when the system explores according to the Linear UCB (LinUCB) strategy [12, 69, 88]. Then we show that the solution can be easily adopted to the simpler problem setting of incentivized exploration without the information gap.

Incentivized exploration under information gap

We present Algorithm 4 to show how the system incentivizes the myopic user to follow the desired exploration strategy under information gap. At each round, the system and the user observe context features $\{\mathbf{v}_a\}_{a\in\mathcal{A}_t}$ and $\{\mathbf{x}_a\}_{a\in\mathcal{A}_t}$ respectively for the same arm set $\mathcal{A}_t$. The system needs to motivate the user to explore arm a_t according to LinUCB strategy based on its current parameter estimation $\hat{\boldsymbol{\theta}}_{v,t}$. To achieve so, the system offers compensation $c_{a_t,t}$ to arm a_t according to Eq (3.13). Note that the system does not offer incentives to the other arms and sets $c_{i,t} = 0, \forall i \neq a_t$. The myopic user pulls the arm that maximizes the sum of his/her estimated reward $\hat{r}_{x,a,t}$ and the compensation $c_{a,t}$. In Lemma 5 we guarantee that the user will pull the system desired arm a_t. Both the system and the user then observe reward feedback r_{a_t}, and update their parameters using ridge regression accordingly.

Denote $\mathit{CB}_{x,t}(\mathbf{x}_a)$ as the width of the user's estimation confidence interval of arm a at time t, which is computed as $\mathit{CB}_{x,t}(\mathbf{x}_a) = \alpha_{x,t}\|\mathbf{x}_a\|_{A_{x,t}^{-1}}$, where $\alpha_{x,t} = R\sqrt{d_x \log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$. $\alpha_{x,t}$ is the upper bound of the width of confidence ellipsoid and is set according to Theorem 2 of [69]. Similar to $\mathit{CB}_{x,t}(\mathbf{x}_a)$, we denote the width of confidence interval on the system side as $\mathit{CB}_{v,t}(\mathbf{v}_a) = \alpha_{v,t}\|\mathbf{v}_a\|_{A_{v,t}^{-1}}$, where $\alpha_{v,t} = R\sqrt{d_v \log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$.

The key challenge in incentivized exploration under information gap is that the system does not maintain the same reward estimation as the user's, because the two sides use different features to learn and predict rewards. This prevents us from computing the minimum required compensation and makes the problem non-trivial. We have to carefully determine the compensation: a larger amount of incentive is required to guarantee that user will explore while we also need to keep the incentives small to maintain a sublinear total compensation. We first use the following lemma to show that on the same arm, the confidence interval by the system's reward estimation is no smaller than the confidence interval by the user's estimate. This lemma guarantees in Algorithm 4 the system provides sufficient incentives to the user to pull its desired arms for exploration.

Lemma 4. *Consider two least square estimators (ridge regression) that estimate the model parameters with the same reward observations but different features satisfying Assumption 1. For all $t \geq 0$ and all arm $a \in \mathcal{A}_t$, we have*

$$\mathit{CB}_{v,t}(\mathbf{v}_a) \geq \mathit{CB}_{x,t}(\mathbf{x}_a), \tag{3.12}$$

i.e., the confidence interval maintained on the system side is no smaller than the user side estimation.

Proof Sketch. Since $\mathit{CB}_{v,t}(\mathbf{v}_a) = \alpha_{v,t}\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}$ and $\mathit{CB}_{x,t}(\mathbf{x}_a) = \alpha_{x,t}\|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$, we can prove $\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}} \geq \|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$ and $\alpha_t^v \geq \alpha_t^x$ separately. It is obvious that $\alpha_t^v \geq \alpha_t^x$ because $d_v \geq d_x$. Substitute $\mathbf{x}_a = P\mathbf{v}_a$ and we can prove that $\mathbf{A}_{v,t}^{-1} - P^\mathsf{T}\left(P\mathbf{A}_{v,t}P^\mathsf{T}\right)^{-1}P$ is a positive semi-definite matrix, which leads to $\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}} \geq \|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$. □

Based on Lemma 4, we have the following lemma.

Lemma 5. *For all $t \geq 0$, with probability at least $1-2\delta$, the users are incentivized to pull the desired arm with compensation*

$$c_{a_t,t} = 4\mathit{CB}_{v,t}(\mathbf{v}_{a_t}) \tag{3.13}$$

to arm

$$a_t = \arg\max_a \left(\mathbf{v}_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{v,t} + 2\mathit{CB}_{v,t}(\mathbf{v}_a)\right), \tag{3.14}$$

i.e., the arm with the highest (relaxed) upper confidence bound according to the system's estimate.

It is worth noting that the system follows a more optimistic arm selection strategy in Eq (3.14) using a confidence interval twice larger than the classical LinUCB algorithm's. We follow this relaxed upper confidence bound because we need to consider the uncertainty on both parties as the first step of the derivation in Eq (3) suggested (in the appendix). It is unclear whether we can incentivize the user to follow the classical LinUCB

Algorithm 5 Incentivized LinUCB without Information Gap

Inputs: λ, δ
Initialize: $\mathbf{A}_x = \lambda \mathbf{I}, \mathbf{b}_x = 0$
for $t = 1$ to T **do**
System and user observe context vectors $\{\mathbf{x}_a\}_{a \in \mathcal{A}_t}$
System calculate compensation $c_{a,t}$ for arm a according to Eq (3.15)
User pulls arm $a_t = \arg\max_{a \in \mathcal{A}} \hat{r}_{x,a,t} + c_{a,t}$
System and user observe reward r_{a_t}
$\mathbf{A}_{x,t+1} \leftarrow \mathbf{A}_{x,t} + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^\intercal$, $\mathbf{b}_{x,t+1} \leftarrow \mathbf{b}_{x,t} + \mathbf{x}_{a_t} r_{a_t}$
$\hat{\boldsymbol{\theta}}_{x,t+1} \leftarrow \mathbf{A}_{x,t+1}^{-1} \mathbf{b}_{x,t+1}$

algorithm. Intuitively, our exploration strategy results in a twice larger regret than the classical LinUCB's, which is still in the same order for T. We provide the regret and compensation upper bound of Algorithm 4 in Section 3.2.4.

Incentivized exploration without information gap

Our solution can be easily adopted to solve the incentivized exploration problem of without information gap, which has not been reported in any existing literature. In Algorithm 5, we show how the system incentivizes the myopic user to follow the desired exploration strategy in this simpler setting.

Without information gap, the system and the user maintain the same parameter and reward estimations, and the *minimum required compensation* to incentivize the user to explore according to LinUCB equals to the difference of the estimated rewards between the currently best arm and the exploratory arm. The system thus only needs to offer compensation by,

$$c_{a_t,t} = \max_i \hat{r}_{x,i,t} - \hat{r}_{x,a_t,t} \tag{3.15}$$

to arm $a_t = \arg\max_a \left(\mathbf{x}_a^\intercal \hat{\boldsymbol{\theta}}_{x,t} + CB_{x,t}(\mathbf{x}_a) \right)$. The user will pull the exploratory arm, because $a_t = \arg\max_i \hat{r}_{x,i,t} + c_{i,t}$, i.e., arm a_t can maximize user's instantaneous utility. Since Algorithm 5 guarantees that the user is incentivized to pull arms according to LinUCB, its regret is in the order of $O(d_x \sqrt{T} \log T)$ as LinUCB (see Theorem 3 of [69]). Its compensation upper bound is stated below.

Theorem 5 (Compensation upper bound without information gap)**.** *With probability at least $1 - \delta$, the total compensation provided in Algorithm 5 is upper bounded as*

$$C(T) \leq \left(R \sqrt{d_x \log \frac{1 + T/\lambda}{\delta}} + \sqrt{\lambda} \right) \sqrt{T d_x \log(\lambda + \frac{T}{d_x})}$$

Proof Sketch. First, with a high probability the compensation at round t is upper bounded by the confidence interval, i.e., $c_{a_t,t} \leq CB_{x,t}(\mathbf{x}_{a_t})$. The total compensation can then be upper bounded by $\sum_t CB_{x,t}(\mathbf{x}_{a_t})$, which can be bounded using Lemma 11 of [69]. □

Note that without information gap, both the regret and compensation upper bounds are in the order of $O(d_x \sqrt{T} \log T)$, with a linear dependency on the feature dimension d_x.

Discussion. Without information gap, i.e., the two parties have access to the same features and maintain the same reward predictions, the system can offer the minimum required compensation as shown in Eq (3.15) to incentivize exploration. With information gap, compensate by Eq (3.13) can still successfully incentivize exploration in a high probability manner, but it is inevitably larger than the minimum amount. More specifically, without information gap the required compensation can be computed deterministically in Eq (3.15); otherwise, the system can only estimate the reward difference with a high probability (as shown in Lemma 5). We also notice without information gap the system does not compensate if the greedy choice also has the largest upper

confidence bound, which happens more often in the later rounds when the reward estimation converges. But with information gap, our algorithm always compensates, because $CB_{v,t}(\mathbf{v}_{a_t}) > 0$, i.e., the system does not know if the user's greedy choice is also preferred in terms of its UCB. We will show in the next section that the total compensation is still sublinear under information gap.

3.2.4 Analysis

We first analyze the regret and compensation upper bound of Algorithm 4. We then discuss the compensation lower bound of the problem.

Regret and compensation upper bound

Theorem 6. *With probability at least $1-3\delta$, the cumulative regret of Algorithm 4 is bounded by*

$$R(T) \leq \left(2R\sqrt{d_v \log \frac{1+T/\lambda}{\delta}} + \sqrt{\lambda}\right)\sqrt{Td_v \log(\lambda + \frac{T}{d_v})}$$

Theorem 10 shows that the cumulative regret of Algorithm 4 is in the order of $O(d_v\sqrt{T\log T})$. The proof mostly follows the regret analysis of LinUCB, though we have to use a wider confidence interval for exploration. Note that the resulting probability is $1-3\delta$, because the users will follow the system's exploration strategy with probability at least $1-2\delta$ as shown in Lemma 5 and the confidence bound holds with probability at least $1-\delta$.

Theorem 7. *With probability at least $1-2\delta$, the total compensation provided in Algorithm 4 is upper bounded by*

$$C(T) \leq \left(4R\sqrt{d_v \log \frac{1+T/\lambda}{\delta}} + \sqrt{\lambda}\right)\sqrt{Td_v \log(\lambda + \frac{T}{d_v})}$$

Theorem 7 shows that the total compensation of Algorithm 4 is in the order of $O(d_v\sqrt{T\log T})$. Combining Theorem 10 and 7, we show that our proposed algorithm can incentivize exploration under information gap and achieve both sublinear regret and compensation. We notice that the two upper bounds linearly depend on the system's feature dimension d_v. Comparing to the no information gap setting where we showed both the regret and compensation is in the order of $O(d_x\sqrt{T\log T})$, the added regret and compensation are $O\big((d_v-d_x)\sqrt{T\log T}\big)$. And the corresponding high probability guarantee drops a little. These results suggest that the complexity/difficulty of the problem is characterized by the dimensionality of the observed context features, which is exactly where the information gap comes from.

Compensation lower bound

We now prove a gap-dependent asymptotic compensation lower bound of incentivized exploration in linear bandits with finite arms, and show that our result recovers the lower bound of incentivized exploration reported in non-contextual bandits in [76].

Let $G_{x,T} = \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{x}_{a_t}\mathbf{x}_{a_t}^\intercal\right]$. Without loss of generality assume arm 1 is the best arm and $\Delta_a = \mathbb{E}[r_1] - \mathbb{E}[r_a] = (\mathbf{x}_1 - \mathbf{x}_a)^\intercal\boldsymbol{\theta}^*$ is the reward gap between arm a and the best arm.

Theorem 8 (Compensation lower bound without information gap)**.** *Consider any consistent algorithm observing context features $\{\mathbf{x}_a\}_{a\in\mathcal{A}}$ that guarantees an $o(T^p)$ regret upper bound for any $T>0$ and $0<p\leq 1$. In order to incentivize a user with a least square estimator of rewards to follow the algorithm's choice, the total compensation $C(T)$ for sufficiently large T is*

$$\Omega\left(c_x(\mathcal{A},\boldsymbol{\theta}^*)\log(T)\right),$$

where $c_x(\mathcal{A}, \boldsymbol{\theta}^*)$ is the optimal value of the following optimization problem

$$\begin{aligned} c_x(\mathcal{A}, \boldsymbol{\theta}^*) &= \inf_{\alpha \geq 0} \sum_{\mathbf{x}_a} \alpha_{\mathbf{x}_a} \frac{\Delta_a}{3} \\ \textit{s.t. } & \|\mathbf{x}_a\|^2_{H^{-1}_{x,T}} \leq \frac{\Delta_a^2}{2}, \forall \mathbf{x}_a \textit{ with } \Delta_a > 0 \end{aligned} \tag{3.16}$$

where $H_{x,T} = \sum_{\mathbf{x}_a} \alpha_{\mathbf{x}_a} \mathbf{x}_{a_t} \mathbf{x}_{a_t}^\mathsf{T}$.

Our proof relies on the following lemmas:

Lemma 6 (Theorem 1 in [89])**.** *Assume $G_{x,T}$ is invertible for sufficiently large T. For all suboptimal $a \in \mathcal{A}$ it holds that*

$$\limsup_{T \to \infty} \log T \|\mathbf{x}_a - \mathbf{x}_1\|^2_{G^{-1}_{x,T}} \leq \frac{\Delta_a^2}{2}$$

Lemma 7 (Theorem 8 in [89])**.** *For any $\delta \in [1/T, 1)$, T sufficiently large and t_0 such that G_{t_0} is almost surely non-singular,*

$$\mathbb{P}\left(\exists t \geq 0, \mathbf{x}_a : |\hat{r}_{x,a,t} - \mathbb{E}[r_a]| \geq \sqrt{\|\mathbf{x}_a\|^2_{G^{-1}_{x,t}} f_{T,\delta}}\right) \leq \delta$$

where for some $c > 0$ universal constant

$$f_{T,\delta} = 2\left(1 + \frac{1}{\log(T)}\right)\log(1/\delta) + cd_x \log(d_x \log(T))$$

Proof Sketch. Suppose an algorithm is consistent with regret $o(T^p)$, Lemma 6 suggests that the algorithm must collect a sufficient number of samples such that the width of the confidence interval is small enough to identify the suboptimal arms. Since the algorithm has $o(T)$ regret, we can find t_1 such that the best arm is pulled at least $T/2$ times; and because of the concentration result in Lemma 7, its confidence interval of the best arm is smaller than $\Delta_2/3$ where Δ_2 is the reward gap between the best arm and second best arm. This means for $t > t_1$ we have $\hat{r}_{x,1,t} \geq \mathbb{E}[r_1] - \Delta_2/3$ with a high probability.

For any other arm a, from Lemma 6 and the concentration bound we can show that it will also be pulled enough times such that its confidence interval is smaller than $\Delta_a/3$ with a high probability after a fixed round t_a. Therefore, for $t > t_a$ we have $\hat{r}_{x,a,t} \leq \mathbb{E}[r_a] + \Delta_a/3$. Combining the two inequalities, we know that after a fixed time point, the minimum required compensation to incentivize the user to pull arm a is $\hat{r}_{x,1,t} - \hat{r}_{x,a,t} \geq \Delta_a/3$. We then solve the optimization problem in Eq (3.16) to obtain the compensation lower bound, i.e., minimize the total compensation while satisfying the consistent constraints that the gaps of all suboptimal arms are identified with high confidence. □

Next, we construct an example to illustrate our lower bound analysis.

Example. When $\{\mathbf{x}_a = e_a \in \mathbb{R}^{d_x}\}_{a \in \mathcal{A}}$ are the basis vectors, the problem reduces to a non-contextual K-armed bandit with $K = d_x$. By setting $\|\mathbf{x}_a\|^2_{H^{-1}_{x,T}} = \Delta_a^2/2$, we have $\alpha_{\mathbf{x}_a} = 2/\Delta_a^2$ and $c_x(\mathcal{A}, \boldsymbol{\theta}^*) = \sum_{a \in \mathcal{A}, \Delta_a > 0} \frac{2}{3\Delta_a}$. This gives us the compensation lower bound as follows,

$$C(T) = \Omega\left(\sum_{a \in \mathcal{A}, \Delta_a > 0} \frac{\log(T)}{\Delta_a}\right)$$

This result recovers the lower bound of incentivized exploration in non-contextual bandits in [76]. We also notice that the result can be further bounded as

$$C(T) = \Omega\left(\frac{d_x \log(T)}{\max_{a \in \mathcal{A}} \Delta_a}\right),$$

where we observe a linear dependency on dimension d_x.

Note that our compensation lower bound has order $\Omega(\log(T))$, because it is gap-dependent. We leave the question of whether one can obtain an $\Omega(\sqrt{T})$ gap-independent compensation lower bound for general infinite arm setting, which will match our upper bound in Theorem 7, as an open problem.

Corollary 1 (Compensation lower bound under information gap)**.** *Consider any consistent algorithm observing context features $\{\mathbf{v}_a\}_{a\in\mathcal{A}}$ that guarantees an $o(T^p)$ regret upper bound for any $T > 0$ and $0 < p \leq 1$. To incentivize the user who observes context features $\{\mathbf{x}_a\}_{a\in\mathcal{A}}$ satisfying Assumption 1 with a least square estimator, the total compensation $C(T)$ for sufficiently large T is*

$$\Omega\left(c_v(\mathcal{A}, \boldsymbol{\theta}^*)\log(T)\right),$$

where $c_v(\mathcal{A}, \boldsymbol{\theta}^*)$ is the optimal value of the following optimisation problem

$$c_v(\mathcal{A}, \boldsymbol{\theta}^*) = \inf_{\alpha\geq 0} \sum_{\mathbf{v}_a} \alpha_{\mathbf{v}_a} \frac{\Delta_a}{3}$$
$$\text{s.t. } \|\mathbf{v}_a\|^2_{H^{-1}_{v,T}} \leq \frac{\Delta_a^2}{2}, \forall \mathbf{v}_a \text{ with } \Delta_a > 0$$

where $H_{v,T} = \sum_{\mathbf{v}_a} \alpha_{\mathbf{v}_a} \mathbf{v}_{a_t} \mathbf{v}_{a_t}^\mathsf{T}$.

The proof of compensation lower bound under information gap mostly follows Theorem 8 by simply replacing the user's feature $\mathbf{x}_a$ with the system's feature $\mathbf{v}_a$. The main difference is that when applying the concentration bound in Lemma 7 to derive the minimum required compensation, we still use $\mathbf{x}_a$ because the minimum amount is based on the *user's* estimated reward difference between the currently best arm and the exploratory arm. However, we notice that $\mathbf{x}_a$ or d_x does not directly appear in this lower bound. The impact of $\mathbf{x}_a$ being in a lower-dimensional space is that we have a faster converging concentration bound to get the confidence interval smaller than $\Delta_a/3$ at an earlier time point. Since we consider $T \to \infty$, this does not change the order of the bound and the final result is dominated by $\mathbf{v}_a$.

Considering a similar example of K-armed bandit setting where $d_v = K$, we can obtain

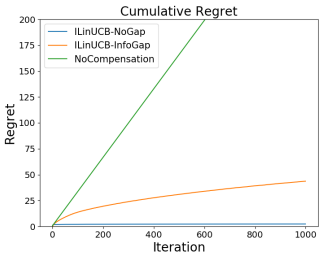
$$C(T) = \Omega\left(\frac{d_v \log(T)}{\max_{a\in\mathcal{A}} \Delta_a}\right)$$

where we observe a linear dependency on dimension d_v.

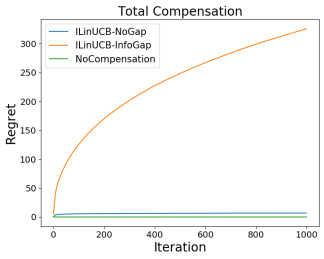
3.2.5 Experiments

We use simulation-based experiments to verify the effectiveness of our proposed incentivized exploration solution. In our simulations, we generate a size-K arm pool $\mathcal{A}$, in which each arm a is associated with a d_v-dimension vector $\mathbf{v}_a$ as the system observed features and a d_x-dimension vector $\mathbf{x}_a$ as the user observed features. Each dimension of $\mathbf{v}_a$ is drawn from a set of zero-mean Gaussian distributions with variances sampled from a uniform distribution $U(0, 1)$. Each $\mathbf{v}_a$ is then normalized to $\|\mathbf{v}_a\|_2 = 1$. We then sample the elements of the $d_x \times d_v$ transformation matrix P from $N(0, 1)$ and normalize each row i by $\|P_i\|_2 = 1$. Following Assumption 1, the user observed features $\mathbf{x}_a$ are generated as $\mathbf{x}_a = P\mathbf{v}_a$. P guarantees that $\|\mathbf{x}_a\|_2 \leq \|\mathbf{v}_a\|_2 = 1$. User's model parameter $\boldsymbol{\theta}^*_x$ is sampled from $N(0, 1)$ and normalized to $\|\boldsymbol{\theta}^*_x\|_2 = 1$. System's model parameter is set to $\boldsymbol{\theta}^*_v = P\boldsymbol{\theta}^*_x$. At each round t, the same set of arms were presented to all the algorithms, but the system and the user observe their different features respectively. After the user pulls an arm a_t, both the user and the system observe its reward following Eq (3.9). We set d_x to 5, d_v to 100, the standard derivation σ of Gaussian noise η_t to 0.1, and the arm pool size K to 100 in our simulations.

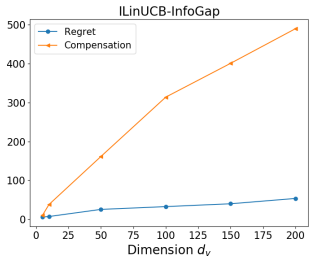
We compare the following algorithms: 1) ILinUCB-InfoGap: our Algorithm 4 where $\{\mathbf{v}_a\}_{a\in\mathcal{A}_t}$ is observed by the system; 2) ILinUCB-NoGap: our Algorithm 5 where both the system and the user observe $\{\mathbf{x}_a\}_{a\in\mathcal{A}}$; 3) NoCompensation: a baseline system that does not offer any compensation to the user. The myopic user always



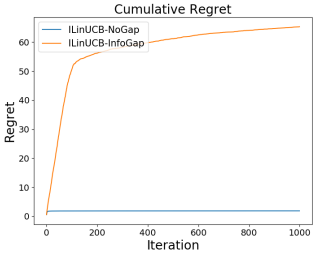
(a) Regret



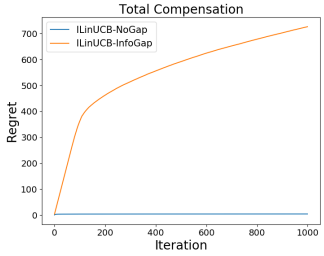
(b) Compensation



(c) Varying dimension d_v



(d) Regret (MAB setting)



(e) Compensation (MAB setting)

Figure 3.3: (a)-(c) Simulation result on randomly sampled features with $d_x = 5$ and $d_v = 100$; (d)-(e) MAB setting where the system only observes the indices of the arms.

pulls the current best arm. We set the probability $\delta = 0.01$ and regularization coefficient $\lambda = 0.1$ for all the algorithms.

We report the averaged results of 10 runs where in each run we sample a random model parameter $\boldsymbol{\theta}_x^*$. In Figure 3.3(a), we observe that without providing any compensation, the myopic user suffers a linear regret, which emphasizes the importance of incentivized exploration. Both ILinUCB-InfoGap and ILinUCB-NoGap enjoy sublinear regret and compensation. The added regret of ILinUCB-InfoGap shows the algorithm explores slower in the large R^{d_v} space because of the information gap.

We notice that the total compensation of ILinUCB-InfoGap in Figure 3.3(b) is sublinear and keeps increasing. The algorithm has to always compensate due to the information gap as we discussed before. ILinUCB-NoGap, however, rarely compensates in the later stage. This is because when system explored sufficiently, greedy choice on the user side agrees with the UCB strategy on the system side, and thus no compensation is needed. In Figure 3.3(c), we vary the dimension of system's feature d_v from 5 to 200 while fixing $d_x = 5$. We observe that both regret and compensation increases linearly with d_v, which confirms our theoretical upper bound.

In Figure 3.3(d) and Figure 3.3(e), we simulate a K-armed bandit setting where only the indices of the arms are available to the system. The system sets $\mathbf{v}_a = e_a \in \mathbb{R}^K$. The rest of the settings are the same as described above. In this setting, our ILinUCB-InfoGap explores almost equivalently to UCB1 [8] and can be viewed as a more optimism version of the Incentivized UCB algorithm in [76] with a wider confidence interval in consideration of the information gap. The system observes the least information in this setting. We notice that its regret and compensation are much larger than the results in Figure 3.3 where $\{\mathbf{v}_a\}_{a\in\mathcal{A}}$ is more informative about the rewards. This again confirms that the system inevitably suffers higher regret and compensation when the features are less informative.

Chapter 4

Privacy and Security in Bandit Learning

The involvement of humans in an interactive learning process brings in both new challenges and opportunities in *privacy and security* perspectives. It is a prominent requirement for intelligent systems to be not only supportive, but also protect the privacy when interacting with humans and be robust to the biased or even adversarial feedback. In this chapter, we introduce our research on privacy and security aspects of bandit learning, aiming to develop interactive systems that are *trustworthy* to humans. We first present the work on utilizing the structural information to balance privacy and utility in a collaborative environment, and then discuss our understanding of the relation between and the vulnerability of linear bandits to data poisoning attack and the geometry of its context features.

4.1 Improving Privacy-utility Trade-off in Collaborative Environments

Personalized recommendation is a double-edged sword: the gained utility also comes with the risk of privacy violation. Overly personalized recommendations could be a potential source of privacy vulnerability, for adversaries to take advantage of, e.g., infer users' sensitive information. Real-world privacy breaches have been reported in Amazon's recommendation system [18] and Facebook's advertisement system [19], where an adversary learns considerable amount of information about a user solely based on the systems' recommendation sequences. Comparing to the offline learnt models, online learning methods directly interact with sensitive user data, e.g., user clicks or purchasing history, and timely update the models to adjust their output, which makes privacy an even more serious concern [90–93]. Realizing its importance, private online learning has recently attracted increasing attention in the research community, with a goal to prevent the algorithm's sequential output from revealing a user's private information. While there is existing research on differentially private online convex optimization [90, 94] and contextual bandits [93, 95], private collaborative bandits have not been explored yet.

The challenges regarding the risk of privacy breach in a collaborative bandit based recommender system are unique. In such a system, the algorithm recommends an item to a user, and the user provides feedback (e.g., click) based on his/her true preference. The feedback (reward) is then used to update not only the model's reward estimation on this user, but also other on users via the imposed dependency among users. As a result, any change in one user's feedback promptly leads to changes in the algorithm's output, e.g., different sequences of recommended items, potentially for *all* users. This is originally designed to improve subsequent recommendations collectively across all users. But a user's private information could thus be inferred and revealed simply by releasing the recommendation sequence, e.g., extraction attack, even if this user's feedback is kept private in the system.

In this work, we propose the first study to equip collaborative bandit algorithms with privacy guarantees, under the notion of *global differential privacy* [96] and *local differential privacy* [97]. Under global differential

privacy, a user is assumed to trust (or say he/she has to trust) the system and provide real engagement data to the system, and the system outputs private recommendations; while under local differential privacy, each user provides perturbed statistics to the system and is no longer required to trust the system or the communication between him/her and system. As the very first study on private collaborative bandits, we focus on algorithms that leverage *known dependency* (e.g., social connections) among users, such as [1, 11]. Specifically, these algorithms propagate the reward collected from one user to update his/her peers' bandit models, according to a given and fixed user dependency structure.

One common practice to achieve privacy guarantee is to inject noise to perturb certain statistics derived from private information in the learning process, either on the server side to achieve global differential privacy or on the client side to achieve local differential privacy [96–98]. However, how to efficiently inject noise in the collaborative bandit learning setting is non-trivial, because of the inherent information sharing mechanism. Specifically, to preserve privacy in collaborative bandits, we apply the tree-based mechanism [99, 100] to add Laplace noise to the models' statistics to guarantee privacy on each user's *reward* feedback (e.g., user clicks). We conduct sensitivity analysis, to which the key is to calibrate the noise scale with respect to the *structure of collaboration* defined by the user dependency graph. Our insight is that a careful sensitivity analysis over the collaboration structure offers the opportunity to inject minimum amount of noise and better balance the *privacy and utility trade-off*. In this work, we employ the collaborative bandit algorithm developed in this dissertation, i.e., Collaborative LinUCB (CoLin) [1], as the baseline algorithm, which represent a classic types of social network based collaboration structure. We develop its private versions to illustrate a general solution framework for private collaborative bandit. We prove the private algorithms reduce the added regret caused by privacy-preserving mechanism compared to its linear bandits counterparts, i.e., collaboration actually helps to achieve stronger privacy with the same amount of injected noise. We also empirically evaluate the algorithms on both synthetic and real-world public datasets to validate its effectiveness and show the improved trade-off between utility and privacy from our proposed solution framework.

4.1.1 Related Work

Differential privacy [96] provides a formal notion to quantify the amount of information an adversary could obtain by observing the algorithm's output. The common practice is to add Laplace or Gaussian noise to the output; and the scale of noise depends on privacy budget (often denoted as ϵ) and *sensitivity*, which is the change of an algorithm's output caused by the change of input. Prior work has studied the problem of differential privacy for offline collaborative filtering methods [101–104].

Differential privacy was first extended to an online setting for stream data in [99, 100]. Differentially private online learning methods have been studied for online convex optimization [90, 91, 105] and bandit problems [92, 93, 95, 106]. The key technique of these solutions is the *tree-based mechanism*, which was proposed in [99, 100] for privately releasing *sum* statistics in stream data with finite time horizon T. Its key idea is to maintain a noisy binary tree where the T leaf nodes are the data points, and the internal nodes in the tree stores the sum of all the leaves in its sub-tree. Each node (which represents a partial sum) in the tree is protected with $\frac{\epsilon}{\log(T)}$-differential privacy. Since each sum statistic can be rewritten into $\lceil \log(T) \rceil$ partial sums, composition theorem of differential privacy [101] guarantees the sequence of output sum statistics is ϵ-differentially private.

Based on this tree-based mechanism, (globally) differentially private linear bandit was first studied in [95] with guaranteed privacy in collected user reward feedback. However, it is non-trivial to extend the private linear bandits to collaborative bandits setting, where one user's reward feedback directly contributes to other users' model update. In other words, the change of model's input from one user can be measured by the model's output in (potentially) all users. This propagation of information has to be carefully reflected in sensitivity analysis to avoid trivial solutions.

4.1.2 Differential Privacy

For a contextual bandit algorithm that interacts with users over time horizon T, denote $S = \{r_t\}_{t=1}^T$ as the reward sequence, where r_t is the reward feedback from user u_t at time t. S' is considered as an adjacent neighboring sequence of S, if it only differs from S at one point of reward r_i. The output of a bandit algorithm $\mathcal{O}$ (which is observed by the adversary) is the sequence of its selected arms, i.e., $\{a_t\}_{t=1}^T$.

Definition 1 (Global Differential Privacy (DP) [96])**.** *A randomized mechanism $\mathcal{M}$ is ϵ-differentially private if for any adjacent neighboring sequences $\{S, S'\}$ and output, $\mathbb{P}\left(\mathcal{M}(S) \in \mathcal{O}\right) \leq e^{\epsilon}\mathbb{P}\left(\mathcal{M}(S') \in \mathcal{O}\right)$.*

Global differential privacy ensures the adversary observes almost the same output from a private algorithm, in a probabilistic sense, if only one input data point is changed. The difference between the corresponding output is characterized by ϵ. Laplace or Gaussian noise is commonly introduced to disguise the output, where the noise scale is related to the privacy budget ϵ and the *sensitivity* of $\mathcal{M}$. We formally define sensitivity below.

Definition 2 (Sensitivity [96])**.** *For any adjacent neighboring sequences $\{S, S'\}$, global sensitivity of a function $f(\cdot)$ is defined as $\Delta_f = \max_{S,S'}|f(S) - f(S')|$.*

Global differential privacy protects sensitive user data from an adversary who has access to the algorithm's output. But it requires the user to send his/her authentic data to the server. Thus, the server and the communication between user and server have to be trusted. To lift the trust needed from the user, local differential privacy (LDP) is proposed [97]. The key idea is that the privacy mechanism needs to perturb the sensitive statistics on the client side before sending it to the server for further computation. Local differential privacy has been adopted in many real-world applications, such as the RAPPOR system developed by Google to collect web browsing behaviour [107], and Apple provides this privacy protection when collecting users' usage and typing history [108]. Note that the input and output of a local differential privacy mechanism could be different from the global differential privacy mechanism, even for the same problem, as they impose different privacy requirements. Let S_i be the reward sequence of user u_i such that $\bigcup_i S_i = S$. The formal definition of local differential privacy is provided below, where a user perturbs his/her private statistics S_i using mechanism L locally, and then send the noisy statistics to the server.

Definition 3 (Local Differential Privacy (LDP) [97])**.** *A randomized mechanism $\mathcal{M}$ is ϵ-locally differentially private if for any input $\{S_i, S_i'\}$ and output $\mathcal{O}$, $\mathbb{P}\left(\mathcal{M}(S_i) \in \mathcal{O}\right) \leq e^{\epsilon}\mathbb{P}\left(\mathcal{M}(S_i') \in \mathcal{O}\right)$*

The key difference between LDP and DP is that a DP mechanism takes all users' data S as input and requires the output to be indifferentiable, while LDP mechanism takes only one user's data S_i as input and generates randomized responses per user (locally) for downstream tasks.

4.1.3 Global Differential Privacy for CoLin

In Collaborative LinUCB (CoLin [1]), contextual bandit models are placed on a weighted graph $G = (\mathcal{V}, \mathcal{E})$, which encodes the affinity relationship among users. Specifically, each node $v_i \in \mathcal{V}$ in G hosts a bandit model parameterized by $\boldsymbol{\theta}_i$ for user i; and the edges in $\mathcal{E}$ represent the affinity relation over pairs of users. This graph is encoded as an $N \times N$ stochastic matrix $\mathbf{W}$, in which each element w_{ij} is nonnegative and proportional to the influence that user i has on user j. $\mathbf{W}$ is normalized such that $\sum_{i=1}^{N} w_{ij} = 1$ for $j \in \{1, ..., N\}$, and it is assumed to be time-invariant and known to the learner beforehand. Accordingly, CoLin postulates an *additive* reward generation assumption: the expected reward $\mathbb{E}[r_{a_t,u_t}]$ is not only determined by user u_t's own preference on the arm a_t, but also by that from the neighbors who have influence on u_t as $\mathbb{E}[r_{a_t,u_t}] = \sum_{j=1}^{N} w_{u_t j}\mathbf{x}_{a_{t,j}}^{\mathsf{T}}\boldsymbol{\theta}_j$; or equivalently this can be described as,

$$r_{a_t,u_t} \sim N\big(\text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})^{\mathsf{T}}\text{Vec}(\boldsymbol{\Theta}), \sigma^2\big) \tag{4.1}$$

where $\text{Vec}(\cdot)$ is the matrix vectorization operation, $\boldsymbol{\Theta}$ is a $d \times N$ matrix consisting of parameters from all the bandits in the graph: $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, and $\mathring{\mathbf{X}}_{a_t,u_t}$ is a $d \times N$ matrix with only the column corresponding to user u_t at time t set to $\mathbf{x}_{a_t,u_t}^{\mathsf{T}}$ and all the other columns set to zero. By defining $\tilde{\mathbf{x}}_{a_t,u_t} = \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})$ and $\boldsymbol{\vartheta} = \text{Vec}(\boldsymbol{\Theta})$, Eq (4.1) can be re-written as $r_{a_t,u_t} \sim N(\tilde{\mathbf{x}}_{a_t,u_t}^{\mathsf{T}}\boldsymbol{\vartheta}, \sigma^2)$.

With such a collaborative reward generation assumption, CoLin appeals to ridge regression for estimating the global bandit parameter matrix $\boldsymbol{\vartheta}_t$ over all the users at time t. It has a closed-form solution $\hat{\boldsymbol{\vartheta}}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$, in which $\mathbf{A}_t = \lambda\mathbf{I}_{dN} + \sum_{t'=1}^{t-1}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}^{\mathsf{T}}$ and $\mathbf{b}_t = \sum_{t'=1}^{t-1}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}r_{a_{t'},u_{t'}}$. $\mathbf{I}_{dN}$ is an identity matrix and λ is the trade-off parameter for the L2 regularization in ridge regression.

Algorithm 6 Differentially Private CoLin (DP-CoLin)

Inputs: $\delta \in \mathbb{R}_+$,$\lambda \in [0,1]$, $\mathbf{W} \in \mathbb{R}^{N\times N}$, Δ
Initialize: $\mathbf{A}_1 \leftarrow \lambda \mathbf{I}_{dN\times dN}$, $\mathbf{b}_1 \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\vartheta}}_1^p \leftarrow \mathbf{A}_1^{-1}\mathbf{b}_1$,
for $t = 1$ to T **do**
  Receive user u_t, observe context vectors, $\mathbf{x}_{a_t,u_t} \in \mathbb{R}^d$ and construct $\tilde{\mathbf{x}}_{a_t,u_t} = \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T})$ for $\forall a \in \mathcal{A}$
  Take action $a_t = \arg\max_{a\in\mathcal{A}} \tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}\hat{\boldsymbol{\vartheta}}_t^p + \alpha_t\sqrt{\tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}\mathbf{A}_t^{-1}\tilde{\mathbf{x}}_{a_t,u_t}}$, where α_t is given by Lemma 9.
  Observe payoff r_{a_t,u_t}
  $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + \tilde{\mathbf{x}}_{a_t,u_t}\tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}$, $\mathbf{b}_{t+1} \leftarrow \mathbf{b}_t + \tilde{\mathbf{x}}_{a_t,u_t}r_{a_t,u_t}$
  Sample noise $\eta_t \sim \text{TreeMechanism}(\Delta,\epsilon)$, in which $\Delta = \max_i L\|\mathbf{W}_i\|_2$
  $\mathbf{b}_{t+1}^p \leftarrow \mathbf{b}_{t+1} + \eta_t$, $\hat{\boldsymbol{\vartheta}}_{t+1}^p \leftarrow \mathbf{A}_{t+1}^{-1}\mathbf{b}_{t+1}^p$

The required information sharing in CoLin brings unique challenges in protecting users' reward feedback, i.e., the change in one user's reward feedback can be effectively inferred from all users' observed recommendation sequences. The recommendation sequences for all users thus have to be perturbed to obtain differential privacy. But instead of directly adding noise to the model's output, i.e., its choice of arms, we choose to add noise η_t to the sufficient statistics $\mathbf{b}_t = \sum_{t'}^{t-1}\tilde{\mathbf{x}}_{a_{t'}}r_{t'}$ in CoLin, where we sample η_t from a tree-based mechanism [99, 100]. Because differential privacy is immune to post-processing [109], this ensures differential privacy on the algorithm's output. We name this private derivation of CoLin as (Globally) Differentially Private CoLin (DP-CoLin), and provide its details in Algorithm 6.

The key in DP-CoLin is to derive the sensitivity of CoLin. Analyzing sensitivity in a linear bandit is straightforward [95], as the sensitivity on $\mathbf{b}_t$ can be directly bounded by $\|\mathbf{x}_a\|_2|r_a - r'_a| \leq L$, where the reward difference is bounded by 1 and the norm of context vector is bounded by L. However, for collaborative bandits, the context vectors encode user dependency and have a higher dimension $\tilde{\mathbf{x}}_{a,u} \in \mathbb{R}^{dN}$. A trivial bound is $\|\tilde{\mathbf{x}}_{a,u}\|_2|r_a - r'_a| \leq NL$; but we argue this is not tight enough and unnecessarily introduces large noise. Below we analyze the privacy guarantee of DP-CoLin with a tighter sensitivity bound, which calibrates the noise with respect to the structure of collaboration embedded in $\mathbf{W}$.

Privacy Analysis of DP-CoLin.

Lemma 8 provides the sensitivity of model statistics $\mathbf{b}_t$ in CoLin, based on which we develop the privacy guarantee of DP-CoLin.

Lemma 8 (Sensitivity of $\mathbf{b}_t$ in CoLin)**.** *Sensitivity of* $\mathbf{b}_t$ *is* $\Delta = \max_i L\|\mathbf{W}_i\|_2$*, where* $\mathbf{W}_i$ *is the* i*-th row of user dependecy matrix* $\mathbf{W}$ *and* L *is the norm of context vector* $\mathbf{x}$*.*

The proof of this lemma is provided in Appendix. Note that the sensitivity Δ of CoLin is related to the structure of $\mathbf{W}$; and we discuss two extreme cases of $\mathbf{W}$ to illustrate its effect on privacy protection. Consider when $\mathbf{W}$ is an identity matrix, the resulting sensitivity by our Lemma 8 is L, which is the same as in linear bandits, since there is no influence among users. When $\mathbf{W}$ is a uniform matrix, i.e., users have homogeneous influence among each other and $w_{ij} = \frac{1}{N}$, Lemma 8 shows the sensitivity is $\frac{L}{\sqrt{N}}$. This result is significant: stronger user dependency in CoLin not only leads to lower regret [1], but also smaller sensitivity of $\mathbf{b}_t$, which directly reduces the level of required noise to guarantee privacy. This result is also intuitive: when every user has uniform influence on each other, it becomes harder to tell whose action causes the observed change in the algorithm's output. Less perturbation is thus needed to protect a single user's privacy. This improvement can hardly be obtained by directly applying existing conclusions on linear bandits.

Based on the above sensitivity analysis, we prove privacy guarantee of DP-CoLin in the following.

Theorem 9 (Privacy of DP-CoLin)**.** *Algorithm 6 with global sensitivity* Δ *defined in Lemma 8 is* ϵ*-differentially private.*

Proof. By applying tree-based mechanism [99, 100] with privacy budget ϵ and sensitivity Δ as shown in line 9-11 of Algorithm 6, the perturbed statistics $\mathbf{b}_t^p$ is ϵ-differentially private. Since differential privacy is immune to post-processing [109], this consequently makes the model parameter $\hat{\boldsymbol{\vartheta}}_t^p$ and the sequence of recommendation $\{a_t : t \in [1..T]\}$ produced by $\hat{\boldsymbol{\vartheta}}_t^p$ also ϵ-differentially private. □

Regret Analysis of DP-CoLin.

We first prove the corresponding confidence bound of parameter estimation in DP-CoLin, i.e., α_t in line 5 of Algorithm 6, which governs its upper confidence bound based arm selection for online learning. In the following discussion, we use $\|\mathbf{B}\|_{\mathbf{A}} = \sqrt{\mathbf{B}^\top \mathbf{A}\mathbf{B}}$ to denote the matrix norm of vector $\mathbf{B}$.

Lemma 9 (Confidence Bound of DP-CoLin)**.** *For any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters in DP-CoLin is bounded by,*

$$\|\hat{\boldsymbol{\vartheta}}_t^p - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\Big) - 2\log(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \frac{\Delta}{\epsilon}\log T\sqrt{\log t}\log\frac{1}{\delta}$$

The proof is provided in Appendix. The right-hand side of the inequality in Lemma 9 gives us α_t that is used in line 5 of Algorithm 6 for arm selection. We notice that in order to maintain a private bandit model $\hat{\boldsymbol{\vartheta}}_t^p$, the parameter estimation error of DP-CoLin suffers from an additional term $\frac{\Delta}{\epsilon}\log T\sqrt{\log t}\log\frac{1}{\delta}$ comparing to that in CoLin due to the added noise η_t. Based on Lemma 9, we have the following theorem about the upper regret bound of the DP-CoLin algorithm, which shows the trade-off between privacy budget ϵ and regret.

Theorem 10 (Regret of DP-CoLin)**.** *With probability at least $1-\delta$, the accumulated regret of DP-CoLin algorithm satisfies,*

$$\begin{aligned}\boldsymbol{R}(T) \leq 2\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_tj}^2}{\lambda dN}\Big)}\Bigg(\sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\Big) - 2\log(\delta)} \\ + \frac{\max_i L\|\mathbf{W}_i\|_2}{\epsilon}\log^{1.5} T\log\frac{1}{\delta}\Bigg)\end{aligned} \tag{4.2}$$

Specifically, the added regret of DP-CoLin comparing to the CoLin is the last term, i.e.,

$$\frac{2\max_i L\|\mathbf{W}_i\|_2}{\epsilon}\log^{1.5} T\log\frac{1}{\delta}\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_tj}^2}{\lambda dN}\Big)}$$

We illustrate the proof details in Appendix. From Theorem 10, we can find that the dependency structure plays an important role in the added regret, and again we discuss those two extreme cases of $\mathbf{W}$ to explain its effect. If $\mathbf{W}$ is an identity matrix, the added regret is in the order of $O\big(\frac{\sqrt{N}}{\epsilon}\log^{1.5} T\sqrt{\log\frac{T}{N}}\sqrt{T}\log\frac{1}{\delta}\big)$. And if $\mathbf{W}$ is a uniform matrix, the added regret is in the order of $O\big(\frac{1}{\epsilon}\log^{1.5} T\sqrt{\log\frac{T}{N}}\sqrt{T}\log\frac{1}{\delta}\big)$. It is important to note that the collaboration structure also helps reduce the added regret, by a factor of $\frac{1}{\sqrt{N}}$, required to achieve privacy. In the meanwhile, the regret reduction from collaboration in the original CoLin is still preserved in the first part of Eq (4.2) in DP-CoLin.

4.1.4 Local Differential Privacy for CoLin

Global differential privacy for CoLin requires each user to send true reward (e.g., clicks) to the server, which then aggregates the data, injects noise, and publishes a privacy preserving output. Local differential privacy lifts the trust on the server by asking each user to perturb his/her data locally, before any disclosure to non-trustful server or the communication. Intuitively, this stronger privacy guarantee is at the cost of worse utility.

We present the Locally Differentially Private CoLin algorithm (LDP-CoLin) in Algorithm 7 in Appendix due to the space limit. LDP-CoLin requires a different communication mechanism: instead of directly sending reward

r_{a_t,u_t} to the server, each user u maintains $\mathbf{b}_{u,t} = \sum_{t'=1}^{t_u-1} \tilde{\mathbf{x}}_{a_{t'},u} r_{a_{t'},u}$ locally as shown in line 8 of Algorithm 7. Each user perturbs their own $\mathbf{b}_{u,t}$ by a tree-based mechanism, where noise scales with per-user sensitivity Δ_u (line 8-9), and then sends it to the server. The server aggregates the received statistics to get $\mathbf{b}_t^p$ as shown in line 12, and uses it for model estimation and subsequent recommendations. Again in LDP-CoLin the key is to analyze the sensitivity, which controls the minimum amount of noise needed for privacy protection.

Algorithm 7 Locally Differentially Private CoLin (LDP-CoLin)

1: **Inputs:** $\delta \in \mathbb{R}_+$,$\lambda \in [0,1]$, $\mathbf{W} \in \mathbb{R}^{N\times N}$, $\Delta_{1:N}$
2: **Initialize:** $\mathbf{A}_1 \leftarrow \lambda \mathbf{I}_{dN\times dN}$, $\mathbf{b}_{u,1} \leftarrow \mathbf{0}$ for $\forall u$, $\hat{\boldsymbol{\vartheta}}_1 \leftarrow \mathbf{A}_1^{-1}\mathbf{b}_1$,
3: **for** $t = 1$ to T **do**
// Sever side:
4: Receive user u_t, observe context vectors $\mathbf{x}_{a_t,u_t} \in \mathbb{R}^d$, and construct $\tilde{\mathbf{x}}_{a_t,u_t} = \text{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^\mathsf{T})$ for $\forall a \in \mathcal{A}$
5: Take action $a_t = \arg\max_{a\in\mathcal{A}} \tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}\hat{\boldsymbol{\vartheta}}_t + \alpha_t\sqrt{\tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}\mathbf{A}_t\tilde{\mathbf{x}}_{a_t,u_t}}$, where α_t is given by Lemma 11.
// User side:
6: Observe r_{a_t,u_t}
7: Update locally $\mathbf{b}_{u_t,t_{u_t}+1} \leftarrow \mathbf{b}_{u_t,t_{u_t}} + \tilde{\mathbf{x}}_{a_t,u_t} r_{a_t,u_t}$
8: Sample noise $\eta_{u_t,t_{u_t}} \sim \text{TreeMechanism}_{u_t}(\Delta_{u_t},\epsilon)$, in which $\Delta_{u_t} = L\|\mathbf{W}_{u_t}\|_2$
9: Send perturbed statistics $\mathbf{b}^p_{u_t,t_{u_t}+1} \leftarrow \mathbf{b}_{u_t,t_{u_t}+1} + \eta_{u_t,t_{u_t}}$ to server
10: $t_{u_t} \leftarrow t_{u_t} + 1$
// Server side:
11: $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + \tilde{\mathbf{x}}_{a_t,u_t}\tilde{\mathbf{x}}_{a_t,u_t}^\mathsf{T}$, $\mathbf{b}^p_{t+1} \leftarrow \sum_u \mathbf{b}^p_{u,t_{u_t}}$, $\hat{\boldsymbol{\vartheta}}^p_{t+1} \leftarrow \mathbf{A}_{t+1}^{-1}\mathbf{b}^p_{t+1}$

Privacy Analysis of LDP-CoLin

We first analyze the sensitivity Δ_u of $\mathbf{b}_{u,t}$ for each user u, and then show that Algorithm 7 is locally differentially private using this per-user sensitivity.

Lemma 10 (Sensitivity of $\mathbf{b}_{u,t}$ in CoLin)**.** *Sensitivity of $\mathbf{b}_{u,t}$ for user u is $\Delta_u = L\|\mathbf{W}_u\|_2$.*

The proof is similar to Lemma 8 and the details are provided in Appendix. The main difference is that sensitivity Δ_u is for a specific user u, which only relies on his/her dependent neighbors, i.e., $\mathbf{W}_u$.

Theorem 11 (Privacy of DP-CoLin)**.** *Randomized response $\mathbf{b}^p_{u,t}$ in Algorithm 7 with sensitivity Δ_u defined in Lemma 10 is ϵ-locally differentially private.*

The proof is similar to DP-CoLin but works in the local setting: as shown in line 8-9 of Algorithm 7, each user u maintains his/her own tree-based mechanism with privacy level ϵ and sensitivity Δ_u *locally*. The local statistics $\mathbf{b}_{u,t}$ are perturbed by the tree-based mechanism thus is ϵ-locally differentially private, and thus are $\hat{\boldsymbol{\vartheta}}^p_t$ and the resulting recommendation sequence.

Regret Analysis of LDP-CoLin

Due to local noise injection, the server's arm selection strategy has to be revised accordingly, which can be guided by the following lemma.

Lemma 11 (Confidence Bound of LDP-CoLin)**.** *Let t_i be the number of times where user i interacts with the system up to time t, i.e., $\sum_i t_i = t$. For any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters in LDP-CoLin is bounded by,*

$$\|\hat{\boldsymbol{\vartheta}}^p_t - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\log\left(1+\frac{\sum_{t'=1}^t\sum_{j=1}^N w^2_{u_{t'}j}}{\lambda dN}\right) - 2\log(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \frac{1}{\epsilon}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^N \log t_i(\Delta_i \log T_i)^2}$$

The proof detail is shown in Appendix. Similarly, the right-hand side of the inequality gives us α_t which is used in line 5 of Algorithm 7. Based on it, we have the following theorem about the upper regret bound of LDP-CoLin.

Theorem 12 (Regret of LDP-CoLin)**.** *With probability at least $1-\delta$, the accumulated regret of LDP-CoLin algorithm (Algorithm 7) satisfies,*

$$\boldsymbol{R}(T) \leq 2\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}\Bigg(\sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'} j}^2}{\lambda dN}\Big) - 2\log(\delta)} \quad (4.3)$$
$$+ \frac{1}{\epsilon}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\|\mathbf{W}_i\|^2\log^3 T_i}\Bigg)$$

Specifically, the added regret of LDP-CoLin comparing to the non-private CoLin is the last term.

Due to space limit, we omit the details of this proof. Note that Theorem 12 is in a general form in which we do not make any assumption about the users' arriving frequency or order. To better illustrate the added regret, we discuss a special case where the frequency of each user interacting with the system is the same, i.e., $T_i = \frac{T}{N}$. The added regret can thus be simplified as,

$$\frac{2}{\epsilon}\log^{1.5}\frac{T}{N}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\|\mathbf{W}_i\|^2}\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}.$$

Consider the best case scenario where $\mathbf{W}$ is a uniform matrix, e.g., maximum collaboration, the added regret in LDP-CoLin is in the order of $O\Big(\frac{\sqrt{N}}{\epsilon}\log^2\frac{T}{N^2}\sqrt{T}\log\frac{1}{\delta}\Big)$, while DP-CoLin only has the added regret of $O\Big(\frac{1}{\epsilon}\log^{1.5}T\sqrt{\log\frac{T}{N^2}}\sqrt{T}\log\frac{1}{\delta}\Big)$. In fact, in both cases of the illustrative dependency structure, e.g., no collaboration and uniform collaboration, the added regret of LDP-CoLin is roughly $\sqrt{N}$-times larger compared with DP-CoLin's, and increases when the number of users grows. This is the inevitable cost to protect privacy in the local (user) level. We verified this relationship between the number of users and regret in our empirical evaluations later as well.

4.1.5 Experiments

We performed empirical evaluations of our developed private collaborative bandit algorithms against several baseline algorithms including the non-private collaborative bandit algorithms CoLin [1] and GOBLin [11], non-private LinUCB [12] and private LinUCB [95]. The datasets include a synthetic dataset from simulation, and two real-world datasets for music recommendation and bookmark recommendation. We compare models' accumulated regret on the synthetic dataset and accumulated reward on real-world datasets.

Evaluation Datasets

• **Synthetic dataset.** To build a synthetic dataset, we follow the settings in [1, 11] to simulate a collaborative online recommendation environment. Specifically, we generate N users, each of which is associated with a d-dimensional parameter vector $\boldsymbol{\theta}^*$, i.e., $\boldsymbol{\Theta}^* = (\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_N^*)$. Each dimension of $\boldsymbol{\theta}_i^*$ is drawn from a uniform distribution $U(0,1)$ and normalized to $\|\boldsymbol{\theta}_i^*\|_2 = 1$. $\boldsymbol{\Theta}^*$ is treated as the ground-truth bandit parameters for reward generation, and they are withheld from bandit algorithms. We construct the golden relational stochastic matrix $\mathbf{W}$ for the graph of users by defining $w_{ij} \propto \langle\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^*\rangle$. We delete the edges where w_{ij} is smaller than a predefined threshold, and get the final user graph G by normalizing each column of $\mathbf{W}$ by its L1 norm. Note that since w_{ij} is generated proportionally to the similarity between $\boldsymbol{\theta}_i^*$ and $\boldsymbol{\theta}_j^*$, the resulting graph naturally satisfies the collaborative assumption in GOBLin [11], i.e., connected users share similar $\boldsymbol{\theta}^*$. The resulting user graph G represented by the relational matrix $\mathbf{W}$ are disclosed to the bandit algorithms. In the end, we generate a size-K arm pool $\mathcal{A}$. Each arm a in $\mathcal{A}$ is associated with a d-dimensional feature vector $\mathbf{x}_a$, each dimension of which is also drawn from $U(0,1)$. We normalize $\mathbf{x}_a$ by its L2 norm.

To simulate the collaborative reward generation process among users, we compute the reward of arm a for user i at time t as $r_{a_{t,i}} = \text{Vec}(\mathring{\mathbf{X}}_{a_t,i}\mathbf{W}^\mathsf{T})^\mathsf{T}\text{Vec}(\mathbf{\Theta}^*) + \gamma_t$ following Eq (4.1), where $\gamma_t \sim N(0, \sigma^2)$. To increase the learning complexity, at each time t, our simulator only discloses a subset of arms in $\mathcal{A}$ to the learning algorithms, e.g., randomly select 10 arms from $\mathcal{A}$ without replacement. In simulation, based on the known bandit parameters $\mathbf{\Theta}^*$, the optimal arm $a^*_{t,i}$ and the corresponding reward $r_{a^*_{t,i}}$ for each user i at time t can be explicitly computed. In our experiment, we set the number of users $N = 10$ and size of arm pool $K = 1,000$. We run $T = 30,000$ iterations and interact with users evenly, which means we serve each user i in total $T_i = 3,000$ iterations.

• **LastFM and Delicious datasets.** These two datasets and the pre-processing of them are the same as that in CoLin.

Experiment Results

• **Regret comparison.** On the synthetic dataset, accumulated regret is used to evaluate the performance of the compared algorithms. In the real-world datasets, since we do not have an oracle policy, we instead use each learning algorithm's accumulated reward for evaluation. The accumulated regret (the lower the better) on the synthetic dataset and accumulated reward (the higher the better) on real-world datasets are reported in Figure 4.1 (a) and Figure 4.2 respectively. We set the privacy budget $\epsilon = 2$ for all private algorithms in our experiments by default.

In both synthetic and real-world datasets, the non-private collaborative bandits performed better than their globally and locally private counterparts, which is surely expected. We also observe that compared with the globally differentially private collaborative bandit algorithms, i.e., DP-CoLin and DP-GOBLin, the locally differentially private algorithms have significantly worse regret (smaller accumulated reward). This is also expected as local differential privacy is a stronger privacy definition on the user side, and more model perturbation has to be introduced to achieve so. Specifically, as our analysis in Section 4.1.4 suggested, the added regret of LDP collaborative bandit algorithms are roughly $\sqrt{N}$-times larger than their DP counterparts.

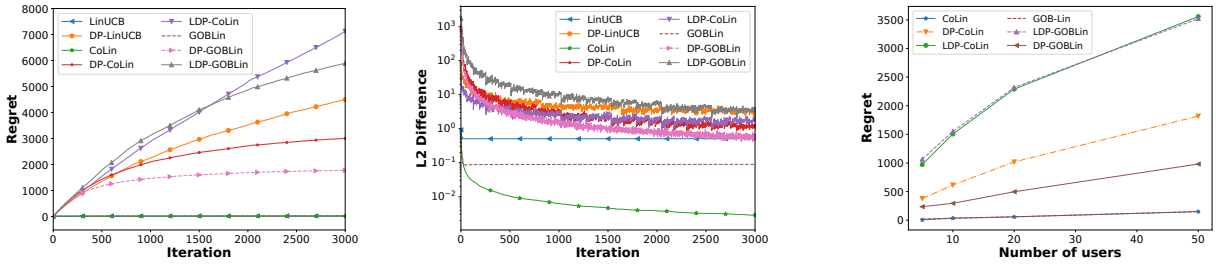
We also notice that DP-CoLin and DP-GOBLin performed better than DP-LinUCB in both synthetic and real-world datasets. The improvement comes from two sources: 1) collaborative learning, which improves the convergence rates of model parameter estimation as discussed in [1, 11]; and 2) privacy mechanism under the collaborative environment, which adds less noise than DP-LinUCB when users are not all independent or disconnected. Accordingly to Figure 4.1 (a), it is obvious that comparing to the regret difference between LinUCB and GOBLin or CoLin, the regret difference between DP-LinUCB and DP-GOBLin or DP-CoLin is much larger. This confirms that the main reason of regret reduction is the calibrated privacy mechanisms developed in this work.

• **Parameter estimation quality.** To better illustrate the performance of different bandit algorithms, we also studied their parameter estimation quality, which directly measures the algorithms' online learning convergence. Specifically, we reported the L2 difference between the estimated bandit parameter $\hat{\boldsymbol{\vartheta}}_t$ and the ground-truth parameter $\boldsymbol{\vartheta}^*$ in Figure 4.1 (b). We observe that private collaborative bandit algorithms have a slower model convergence than their non-private counterparts. Moreover, local differential privacy clearly imposes a much larger estimation error comparing to their counterparts with global differential privacy (note that the y-axis is on a log-scale), which further confirms the required cost to guarantee privacy in the local setting.

Detailed Algorithm-level Analysis

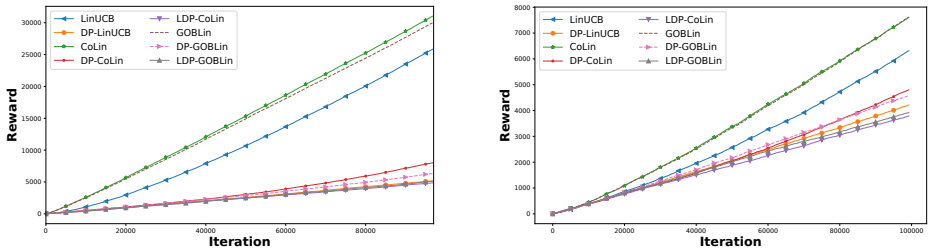
To better understand the trade-off between privacy and utility in collaborative bandit learning, we varied the privacy parameter ϵ and number of users in our evaluation.

• **Effect of privacy budget ϵ.** In Table 4.1, we reported the accumulated regret of the collaborative bandit algorithms with global and local differential privacy under different privacy parameter ϵ. We vary ϵ from 0.5 to 10. We run each experiment for $T = 10,000$ iterations and report the average regret of 5 repeated runs. From the results, we notice a clear trade-off between the required privacy level ϵ and the resulting regret. Stronger privacy requirement (i.e., a smaller ϵ) requires the privacy mechanism to introduce more noise, which



(a) Cumulative regret (b) Parameter estimation error (c) Regret under different number of users

Figure 4.1: Experimental results on synthetic dataset.



(a) Cumulative reward on LastFM dataset. (b) Cumulative reward on Delicious dataset.

Figure 4.2: Experimental results on real-world datasets.

directly inflates regret. This result also supports our theoretical analysis that the added regret of the private collaborative bandit algorithms is in the order of $O(\frac{1}{\epsilon})$.

• **Effect of number of users N.** In Figure 4.1 (c), we show the accumulated regret of the collaborative bandit algorithms with global and local differential privacy under different number of users N. We run $T = 10,000$ iterations and all users are evenly served for $\frac{T}{N}$ times. We vary N from 5 to 50. From the result we observe that the regret increases with the number of users. By looking at the difference between the regret of non-private algorithms and their private versions, we can notice that the added regret increases with number of users N. This also validates our theoretical analysis that the added regret for LDP collaborative bandit algorithms is roughly $\sqrt{N}$ times larger than their DP versions, which is the inevitable cost to protect privacy at the local level.

4.2 Data Poisoning Attack on Linear Bandits

Since bandit algorithms adapt their behavior according to feedback from the environment, it makes such algorithms susceptible to adversarial attacks, especially the data poisoning attacks. Under such an attack, a malicious adversary observes the pulled arm and its reward feedback, and then modifies the reward to misguide the bandit algorithm to pull a target arm, which is of the adversary's interest. Due to the wide applicability of bandit algorithms in practice, understanding the robustness of bandit algorithms under data poisoning attacks becomes increasingly important [21, 22, 110]. However, most known results for data poisoning attacks concentrate on supervised learning settings [111, 112]; so far, little is known about its impact on contextual bandit learning. It is even unknown whether bandit algorithms are vulnerable to data poisoning attacks in general contextual environments.

Recent studies on adversarial attacks in bandits have been focused mainly on context-free settings. Jun et al. [21] and Liu et al. [22] showed that in a stochastic context-free multi-armed bandit environment, an adversary can force any bandit algorithm to pull a target arm linear times only using a logarithmic cost. Garcelon et al. [110] studied poisoning attacks on k-armed linear contextual bandits and showed that linear contextual bandit algorithms can be attacked. Linear stochastic bandits lie in between context-free stochastic bandits and linear contextual bandits. Interestingly, the analysis of its attacks turns out to be arguably more difficult than both, due to arm correlations as we will elaborate later. To our knowledge, there is no unknown result about the attack on linear stochastic bandits.

Table 4.1: Cumulative regret across different bandit algorithm under different privacy level ϵ.

ϵ	0.5	1	2	5	10
DP-LinUCB	3082.90±82.69	2683.69±89.74	1504.14±30.40	910.97±20.83	496.11±14.72
DP-CoLin	2619.56±29.44	2450.70±50.51	1327.70±23.79	884.19±23.12	297.18±6.80
DP-GOBLin	2672.56±29.13	2550.22±19.67	964.63±13.61	685.65±6.47	246.92±9.70
LDP-CoLin	3310.53±51.85	3095.10±48.97	2389.05±61.24	1795.40±31.21	938.76±26.16
LDP-GOBLin	3268.70±65.61	3004.80±75.08	2334.62±63.78	1743.57±36.40	1060.53±28.99

In this work, we study adversarial data poisoning attacks on k-armed linear stochastic bandits, where an adversary modifies the reward using a sublinear budget to misguide the bandit algorithm to pull a target arm $\tilde{x}$ linear times. We answer the following two important research questions: 1) whether a linear stochastic bandit environment is efficiently attackable[1], and 2) if Yes, how can the adversary design an effective attack. Regarding the attackability of an environment, we characterize its nature as the feasibility of a set of linear constraints based on the target arm $\tilde{x}$, all non-target arms $\{x_i\}_{i=1}^k \setminus \tilde{x}$, and the underlying bandit parameter $\boldsymbol{\theta}^*$. The key insight is that attackability is equivalent as finding a parameter vector $\tilde{\boldsymbol{\theta}}$, under which the rewards of all non-target arms are smaller than the reward of target arm $\tilde{x}$ while the reward of $\tilde{x}$ remains the same as that in the original environment specified by $\boldsymbol{\theta}^*$. We prove the feasibility of the constraints is both *sufficient* and *necessary* for attacking a linear stochastic bandit environment. Intuitively, to effectively promote the target arm $\tilde{x}$, an adversary needs to lower the rewards of non-target arms in the *null space* of $\tilde{x}$ by $\tilde{\boldsymbol{\theta}}$.

Inspired by the attackability analysis, we propose a two-stage attack framework against linear stochastic bandit algorithms and demonstrate its application against LinUCB [12] and Robust Phase Elimination [113]: the former is one of the most widely used linear contextual bandit algorithms, and the latter is a robust version designed for settings with adversarial corruptions. In the first stage, our method collects a carefully calibrated amount of rewards on the target arm to assess whether the given environment is attackable. The decision is based on an "empirical" version of our feasibility characterization. If attackable, i.e., there exists a feasible solution $\tilde{\boldsymbol{\theta}}$, in the second stage the method depresses the rewards the bandit algorithm receives on non-target arms based on $\tilde{\boldsymbol{\theta}}$, in order to fool the bandit algorithm to recognize the target arm as optimal.

4.2.1 Preliminaries

Linear stochastic bandit. We study the problem of adversarial attacks on k-arm linear stochastic bandit, where a bandit algorithm sequentially interacts with an environment for T rounds. In each round t, the algorithm pulls an arm a_t from a set $\mathcal{A} = \{x_i\}_{i=1}^k$ with k arms, and receives reward r_t from the environment. Each arm a is associated with a d-dimension context feature vector $x_a \in \mathbb{R}^d$ and we assume $\|x_a\|_2 \leq 1$. The expected reward of arm a is assumed to be a linear function of both context feature and unknown bandit parameter $\boldsymbol{\theta}^*$, i.e., $\mathbb{E}[r_a] = x_a^\mathsf{T}\boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^* \in \mathbb{R}^d$ and we assume $\|\boldsymbol{\theta}^*\|_2 \leq 1$. After pulling arm a_t, the algorithm observes reward feedback $r_{a_t,t} = x_{a_t}^\mathsf{T}\boldsymbol{\theta}^* + \eta_t$, where η_t is an R-sub-Gaussian noise term. The performance of a bandit algorithm is evaluated by its pseudo-regret, which is defined as $R(T) = \sum_{t=1}^T (x_{a^*}^\mathsf{T}\boldsymbol{\theta}^* - x_{a_t}^\mathsf{T}\boldsymbol{\theta}^*)$, where a^* is the best arm according to $\boldsymbol{\theta}^*$, i.e., $x_{a^*} = \arg\max_{x\in\mathcal{A}}[x^\mathsf{T}\boldsymbol{\theta}^*]$. Due to the possible correlation among the context vectors, manipulating the reward of an arm will also change the reward estimation of other correlated arms. This is different from the k-arm linear contextual bandits setting considered in [110], where each arm has an unknown bandit parameter and the reward estimation is independent among arms. Thus the reward manipulation of an arm will not affect other arms.

LinUCB. LinUCB [12, 69] is a classical algorithm for linear stochastic bandit. It estimates a bandit model parameter $\hat{\boldsymbol{\theta}}$ using ridge regression, i.e., $\hat{\boldsymbol{\theta}}_t = \mathbf{A}_t^{-1}\sum_{i=1}^t x_{a_i} r_i$, where $\mathbf{A}_t = \sum_{i=1}^t x_{a_i}x_{a_i}^\mathsf{T} + \lambda\mathbf{I}$ and λ is the coefficient of L2-regularization. We use $\|x\|_\mathbf{A} = \sqrt{x^\mathsf{T}\mathbf{A}x}$ to denote the matrix norm of vector x. Confidence bound about reward estimation on arm x is defined as $\text{CB}_t(x) = \alpha_t\|x\|_{\mathbf{A}_t^{-1}}$, where α_t is a high probability bound of $\|\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_t\|_{\mathbf{A}_t}$. In each round t, LinUCB pulls an arm with the highest upper confidence bound, i.e.,

[1] "Efficient attack" in this work means fooling bandit algorithm to pull the target arm for linear times with sublinear attack cost. We use *attackable* and *efficiently attackable* interchangeable, as the adversary normally only has a limited budget and needs to design a cost-efficient strategy.

$a_t = \arg\max_a [x_a^\intercal \hat{\boldsymbol{\theta}}_t + \text{CB}_t(x_a)]$ to balance the explore-exploit trade-off. LinUCB algorithm achieves a sublinear upper regret bound [69, 88], i.e., $R(T) = \tilde{O}(\sqrt{T})$ ignoring logarithmic term.

Threat model. The goal of an adversary is to fool the bandit algorithm to pull the target arm $\tilde{x} \in \mathcal{A}$ for $T - o(T)$ times. Like most recent works in this space [21, 22, 110, 114], we consider the data poisoning attack on the rewards: after arm a_t is pulled by the bandit algorithm, the adversary modifies the original reward r_{a_t} from the environment by Δr_t to become $\tilde{r}_{a_t}$, i.e., $\tilde{r}_{a_t} = r_{a_t} + \Delta r_t$, and provides the manipulated reward $\tilde{r}_{a_t}$ to the algorithm. Naturally, the adversary should achieve its attack goal with minimum attack cost $C(T) = \sum_{t=1}^{T} |\mathbb{E}[\Delta r_t]|$. An attack strategy is considered *efficient*, if it achieves a sublinear cost, i.e., $C(T) = o(T)$. Note that the expectation of reward manipulation Δr_t is taken with respect to only the sub-Gaussian noise in the rewards.

4.2.2 The Attackability of Linear Stochastic Bandits

The main goal of this work is to study the attackability of a linear stochastic bandit *environment*. At a first glance, one might wonder whether *attackability* should be a property of bandit *algorithms* rather than a property of the environment, since if an algorithm can be attacked, we should have "blamed" the algorithm for not being robust enough, rather than blaming the environment. A key insight of this work is that *attackability is also a property of the linear stochastic bandit environment*.

Definition 4 (Attackability of a k-Arm Linear Stochastic Bandit Environment)**.** *A k-arm linear stochastic bandit environment $\langle \mathcal{A} = \{x_i\}_{i=1}^k, \boldsymbol{\theta}^* \rangle$ is attackable with respect to (w.r.t.) target arm $\tilde{x} \in \mathcal{A}$ if for any no-regret learning algorithm, there exists an attack method that uses $o(T)$ attacking budget but fools the algorithm to pull arm $\tilde{x}$ at least $T - o(T)$ times for any T large enough.*

It is worthwhile to further digest the above definition of attackability. First, this definition is all about the bandit environment $\langle \mathcal{A}, \boldsymbol{\theta}^* \rangle$ and the target arm $\tilde{x}$, but independent of any specific bandit algorithms. Second, if an attack method can only fool a bandit algorithm to pull the target arm $\tilde{x}$ under (only) a few different time horizons T, it does not count as a successful attack — a successful attack has to succeed for infinitely many time horizons. Third, by reversing the order of quantifiers, we obtain an assertion that a bandit environment is not attackable w.r.t. the target arm $\tilde{x}$ if *there exists some no-regret learning algorithm* such that no attack method can use $o(T)$ attack budget to fool the algorithm to pull arm $\tilde{x}$ at least $T - o(T)$ times for any T large enough. The following simple yet insightful example illustrates that there are indeed linear stochastic bandit setups in which some no-regret learning algorithm *cannot* be attacked.

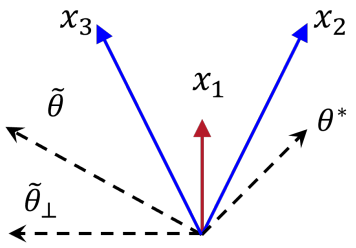


Figure 4.3: Illustration of attackability.

Example 1 (An unattackable environment)**.** *Figure 4.3 shows a three-arm environment with $\mathcal{A} = \{x_1 = (0, 1), x_2 = (1, 2), x_3 = (-1, 2)\}$. Suppose the target arm $\tilde{x} = x_1$ and the ground-truth bandit parameter $\boldsymbol{\theta}^* = (1, 1)$. The expected true rewards of the arms are $r_1^* = 1, r_2^* = 3, r_3^* = 1$ and x_2 is the best arm in this environment. We give an intuitive argument here that this environment with target arm $\tilde{x}$ is not attackable, while its formal proof is an instantiation of our Theorem 13. Specifically, we argue that LinUCB cannot be attacked in the above environment (our argument shall generalize to any linear-regression based no-regret algorithms). Suppose, for the sake of contradiction, that there exists an efficient attack which fools LinUCB to pull x_1 $T - o(T)$ times. Therefore, LinUCB must estimate $\boldsymbol{\theta}^*$ in the x_1's direction almost accurately as T becomes large, since the $\Omega(T)$ amount of true reward feedback in x_1 direction will ultimately dominate the $o(T)$ adversarial contamination. This suggests that the estimated parameter $\hat{\boldsymbol{\theta}}_t$ will be close to $\to (\alpha, 1)$ for some α. Since $(\alpha, 1)^\intercal (x_2 + x_3) = 4$, implying that either x_2 or x_3 will have its estimated reward larger than 2 (i.e., strictly larger than x_1's reward) for any α. This shows that x_1 cannot be the best arm in LinUCB's estimation, which causes a contradiction. therefore, we can safely conclude that this environment cannot be attacked given $o(T)$ budget.*

Note that when $\mathcal{A} = \{x_1, x_2\}$, the environment becomes attackable: as shown in the figure, a feasible attack strategy is to perturb reward according to $\tilde{\boldsymbol{\theta}} = (-2, 1)$. The key idea is that in the null space of x_1, $\tilde{\boldsymbol{\theta}}_\perp$ reduces

the reward of x_2 to make x_1 the best arm but without changing the reward of x_1 from the environment. The presence of arm x_3 prevents the existence of such $\tilde{\boldsymbol{\theta}}_\perp$ and makes the environment unattackable.

The above example motivates us to study when a linear stochastic bandit environment is attackable. After all, if we are facing an unattackable environment, we could safely design no-regret algorithms without any concern of some particular type of adversarial attacks. As Example 1 shows, the attackability of a bandit algorithm depends on the arm set $\mathcal{A} = \{x_i\}_{i=1}^k$, the target arm $\tilde{x}$, and the underlying bandit parameter $\boldsymbol{\theta}^*$.

We now proceed to give a complete characterization about what bandit environments are attackable. For clarity of presentation, in this section, we shall always assume that the adversary knows exactly the ground-truth bandit parameter $\boldsymbol{\theta}^*$ and thus the true expected reward of each arm. This is also called the *oracle attack* in previous works [21, 22, 115]. However, in the next section, we will show that this assumption is not needed: when the bandit environment is indeed attackable, we can design provably successful attacks even if the adversary does not know the underlying bandit parameter $\boldsymbol{\theta}^*$.

We need the following convenient notation to state our result. Let

$$\boldsymbol{\theta}^*_{\parallel} = \frac{\tilde{x}^\mathsf{T}\boldsymbol{\theta}^*}{\|\tilde{x}\|_2^2}\tilde{x} \tag{4.4}$$

denote the projection of ground-truth bandit parameter $\boldsymbol{\theta}^*$ onto $\tilde{x}$ direction. Since the attackability depends on the target arm $\tilde{x}$ as well, we shall include the target arm $\tilde{x}$ as part of the bandit environment. The following theorem provides a clean characterization of attackability.

Theorem 13 (Characterization of Attackability)**.** *A bandit environment $\langle \mathcal{A} = \{x_a\}, \boldsymbol{\theta}^*, \tilde{x}\rangle$ is attackable* if and only if *the optimal objective ϵ^* of the following linear program satisfies $\epsilon^* > 0$.*

$$\begin{array}{ll} \textit{maximize} & \epsilon \\ \textit{subject to} & \tilde{x}^\mathsf{T}\boldsymbol{\theta}^*_{\parallel} \geq \epsilon + x_a^\mathsf{T}(\boldsymbol{\theta}^*_{\parallel} + \tilde{\boldsymbol{\theta}}_\perp), \quad \textit{for } x_a \neq \tilde{x}. \\ & \tilde{x}^\mathsf{T}\tilde{\boldsymbol{\theta}}_\perp = 0 \\ & \epsilon \leq 1 \end{array} \tag{4.5}$$

where $\epsilon \in \mathbb{R}$ and $\tilde{\boldsymbol{\theta}}_\perp \in \mathbb{R}^d$ are variables.

Since LP (4.5) and its solutions will show up very often in our later discussions, we provide the following definition for reference convenience.

Definition 5 (Attackability Index and Certificate)**.** *The optimal objective ϵ^* of LP* (4.5) *is called the attackability index and the optimal solution $\tilde{\boldsymbol{\theta}}_\perp$ to LP* (4.5) *is called the attackability certificate.* [2]

Notably, both the index ϵ^* and certificate $\tilde{\boldsymbol{\theta}}_\perp$ are intrinsic to the bandit environment $\langle \mathcal{A} = \{x_a\}, \boldsymbol{\theta}^*, \tilde{x}\rangle$, and are irrelevant to any bandit algorithms used. As we will see in the next section when designing attack algorithms *without* the knowledge of $\boldsymbol{\theta}^*$, the attackability index ϵ^* will determine how difficult it is to attack the environment.

Proof of Theorem 13. **Proof of sufficiency.** Suppose the attackability index $\epsilon^* > 0$ and let $\tilde{\boldsymbol{\theta}}_\perp$ be a certificate. We design the **oracle null space attack** based on the knowledge of bandit parameter $\boldsymbol{\theta}^*$. Let $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^*_{\parallel} + \tilde{\boldsymbol{\theta}}_\perp$ where $\boldsymbol{\theta}^*_{\parallel}$ is defined in Eq (4.4). The adversary perturbs the reward of any non-target arm $x_a \neq \tilde{x}$ as $\tilde{r}_{a,t} = x_a^\mathsf{T}\tilde{\boldsymbol{\theta}} + \tilde{\eta}_t$, whereas the reward of the target arm $\tilde{x}$ is *not* touched. In other words, the adversary misleads the algorithm to believe that $\tilde{\boldsymbol{\theta}}$ is the ground-truth parameter. To make attack appear less "suspicious", a sub-Gaussian noise $\tilde{\eta}_t$ is added to the perturbed reward to make it stochastic. The key idea is that the attacker does not need to perturb the reward of target arm because $\tilde{x}^\mathsf{T}\tilde{\boldsymbol{\theta}} = \tilde{x}^\mathsf{T}\boldsymbol{\theta}^*_{\parallel} = \tilde{x}^\mathsf{T}\boldsymbol{\theta}^*$, i.e., the original reward is the same as perturbed reward in expectation. Instead, the attacker only perturbs the reward in the *null space* of $\tilde{x}$ according to $\tilde{\boldsymbol{\theta}}_\perp$, which guarantees the cost-efficiency of the attack. Details of the attack can be found in appendix.

[2] We may omit "attackability" when it is clear from the context, and simply mention *index* and *certificate*.

Since the perturbed rewards observed by the bandit algorithm are generated by $\tilde{\boldsymbol{\theta}}$, target arm $\tilde{x}$ is the optimal arm in this environment due to the attackability index ϵ^* being strictly positive. Any no-regret bandit algorithm will only pull the other non-target arms $o(T)$ times and pull target arm $T - o(T)$ times. Thus the attack is successful. Moreover, the cost of oracle attack is $o(T)$ because the attacker only perturbs rewards on the non-target arms for $o(T)$ times, and the expected cost on each attack is bounded by a constant (because of the finite norm of x_a and $\boldsymbol{\theta}^*$).

Proof of necessity. We discuss the proof sketch here and leave the detailed proof in the appendix. Specifically, we shall prove that if $\epsilon^* \leq 0$, the bandit environment is not attackable. To prove this, we will need to identify at least one no-regret bandit algorithm such that no attack strategy can succeed in attacking it. In particular, we will show that LinUCB is robust to any attack strategy with $o(T)$ budget when $\epsilon^* \leq 0$.

LinUCB maintains a model estimate $\hat{\boldsymbol{\theta}}_t$ at round t using the attacked rewards $\{\tilde{r}_{1:t}\}$. We decompose $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t,\parallel} + \hat{\boldsymbol{\theta}}_{t,\perp}$, where $\tilde{x} \perp \hat{\boldsymbol{\theta}}_{t,\perp}$ and $\tilde{x} \parallel \hat{\boldsymbol{\theta}}_{t,\parallel}$. Suppose, for the sake of contradiction, LinUCB is attackable assuming $\epsilon^* \leq 0$. According to Definition 4, the target arm $\tilde{x}$ will be pulled $T - o(T)$ times for infinitely many different time horizons T. Note that the following inequalities hold when $\tilde{x}$ has the unique largest UCB score (and thus is pulled with probability 1):

$$\tilde{x}^\mathsf{T}\hat{\boldsymbol{\theta}}_{t,\parallel} + \mathrm{CB}_t(\tilde{x}) > x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{t,\parallel} + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{t,\perp} + \mathrm{CB}_t(x_a), \forall x_a \neq \tilde{x} \tag{4.6}$$

By attackability, we know that the above inequality will hold for infinitely many t's. As $t \to \infty$, we have $\mathrm{CB}_t(\tilde{x}) \to 0$, and $\mathrm{CB}_t(x_a)$ is strictly positive. Moreover, the estimation of $\hat{\boldsymbol{\theta}}_{t,\parallel}$ will converge to $\boldsymbol{\theta}^*_\parallel$ since $\tilde{x}$ will be pulled for $t - o(t)$ times. By letting $t \to \infty$ in both sides of the above inequalities, we have the following conclusion:

$$\tilde{x}^\mathsf{T}\boldsymbol{\theta}^*_\parallel > x_a^\mathsf{T}\boldsymbol{\theta}^*_\parallel + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{t,\perp}, \forall x_a \neq \tilde{x} \tag{4.7}$$

This implies that there must exist a $\hat{\boldsymbol{\theta}}_{t,\perp}$ that satisfies $\tilde{x} \perp \hat{\boldsymbol{\theta}}_{t,\perp}$ and makes the objective of LP (4.5) strictly above 0. Therefore, its optimal objective ϵ^* must also be strictly positive. This however contradicts our assumption $\epsilon^* \leq 0$, implying that LinUCB is not attackable by any attack strategy. □

We now provide an intuitive explanation about Theorem 13. LP (4.5) is to find a vector $\tilde{\boldsymbol{\theta}}_\perp$ such that: 1) it is orthogonal to $\tilde{x}$ (hence its subscript); and 2) it maximizes the gap ϵ between $\tilde{x}^\mathsf{T}\boldsymbol{\theta}^*_\parallel$ and the largest $x_a^\mathsf{T}(\boldsymbol{\theta}^*_\parallel + \tilde{\boldsymbol{\theta}}_\perp)$ among all $x_a \neq \tilde{x}$. Theorem 13 states that the bandit environment is attackable *if and only if* such a gap (i.e., the attackability index) is strictly larger than 0, i.e., when the *geometry of arm features* allows the adversary to lower non-target arms' rewards by attacking in the null space of $\tilde{x}$. The constraint $\epsilon^* \leq 1$ is to ensure the cost of each attack is bounded by a constant.

Recent works have shown that any no-regret learning algorithm for the context-free k-armed setting (where arm set $\mathcal{A}$ is orthonormal) can be attacked [22] — i.e., a stochastic bandit environment is attackable under our definition. This finding turns out to be a corollary of Theorem 13.

Corollary 2. *For standard stochastic bandit setting where arm set $\mathcal{A}$ is orthonormal, the environment $\langle \mathcal{A} = \{x_a\}, \boldsymbol{\theta}^*, \tilde{x}\rangle$ is attackable for any target arm $\tilde{x}$.*

Proof. Since arms are orthogonal to each other, it is easy to see that $\tilde{\boldsymbol{\theta}}_\perp = -C[\sum_{x_a : x_a \neq \tilde{x}} x_a]$ achieves objective value $C - \tilde{x}^T\boldsymbol{\theta}^*_\parallel$ in LP (4.5). Letting C be a large enough positive constant such that the objective value is positive gives us a feasible $\tilde{\boldsymbol{\theta}}_\perp$ to LP (4.5), which yields the corollary. □

The intuition behind this corollary is that arms in context-free stochastic bandits are independent, and an adversary can lower the rewards of non-target arms to make the target arm optimal. This is also the attack strategy in [21, 22]. Garcelon et al. [110] showed that similar idea works for k-arm linear contextual bandits, where each arm is associated with an unknown bandit parameter and the reward estimations are independent among different arms. Arguably, our setting is more challenging since arms are correlated and the simple attack idea may not be successful as shown in our Example 1. Our analysis characterizes the attackability

Algorithm 8 Two-stage Null Space Attack

1: **Inputs:** T, T_1
2: **Initialize:**
3: Compute $\boldsymbol{\theta}_0 = \arg\max_{\|\boldsymbol{\theta}\|_2 \le 1} \left[\tilde{x}^T \boldsymbol{\theta} - \max_{x_a \neq \tilde{x}} x_a^T \boldsymbol{\theta}\right]$ and let ϵ_0^* be its optimal objective
4: **if** $\epsilon_0^* \le 0$ **then** ▷ Initial attackability test
5: **return** Not attackable and stop.
6: **for** $t = 1$ to T_1 **do** ▷ Attack stage
7: Set $\tilde{r}_t = x_{a_t}^\mathsf{T} \boldsymbol{\theta}_0 + \tilde{\eta}_t$ ▷ Always attack as $\tilde{x}$ is the best
8: Bandit algorithm observes modified reward $\tilde{r}_t$
9: Estimate $\tilde{\boldsymbol{\theta}}_\| = \frac{\sum_{i=1}^{n(\tilde{x})} r_i(\tilde{x})}{n(\tilde{x})\|\tilde{x}\|_2^2}\tilde{x}$
10: Solve LP (4.5) using $\tilde{\boldsymbol{\theta}}_\|$ to obtain estimated attackability index $\tilde{\epsilon}^*$ and certificate $\tilde{\boldsymbol{\theta}}_\perp$
11: **if** $\tilde{\epsilon}^* \le 0$ **then** ▷ Attackability test
12: **return** Not attackable, and stop
13: **else** ▷ Attack stage
14: Set $\tilde{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}}_\| + \tilde{\boldsymbol{\theta}}_\perp$
15: **for** $t = T_1 + 1$ to T **do**
16: **if** $x_{a_t} = \tilde{x}$ for the first time **then** ▷ Compensate $\tilde{x}$
17: Set $\tilde{r}_t = n(\tilde{x}) \times \tilde{x}^\mathsf{T}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) + \tilde{x}^\mathsf{T}\tilde{\boldsymbol{\theta}} + \tilde{\eta}_t$
18: **else**
19: Set $\tilde{r}_t = x_{a_t}^\mathsf{T}\tilde{\boldsymbol{\theta}} + \tilde{\eta}_t$
20: Bandit algorithm observes modified reward $\tilde{r}_t$

based on the geometry of arm features: when the geometry forbids an adversary from lowering the rewards of non-target arms in the null space of the target arm, the environment is unattackable.

4.2.3 Effective Attacks Without Knowledge of True Model Parameters

In the previous section, we characterized the attackability of a linear stochastic bandit environment by assuming the knowledge of ground-truth bandit parameter $\boldsymbol{\theta}^*$. We now show that such oracle knowledge is actually not needed when designing executable attacks. Towards this end, we propose provably effective attacks against two representative bandit algorithms, the most well-known LinUCB [69] and a state-of-the-art robust linear stochastic bandit algorithm based on robust phase elimination [113]. Their different level of robustness turns out to lead to different amount of required attack cost, which further illustrates that the attack analysis often goes hand-in-hand with robustness analysis.

Two-stage Null Space Attack. Our proposed attack method is presented in Algorithm 8. The adversary spends the first T_1 rounds as the first stage to attack rewards on all the arms by imitating a bandit environment $\boldsymbol{\theta}_0$, in which $\tilde{x}$ is the best arm such that arm $\tilde{x}$ will be pulled most often by the bandit algorithm. This stage is for the adversary to observe the rewards for $\tilde{x}$ from the environment, which helps it estimate the parameter $\boldsymbol{\theta}_\|^*$. At round T_1, the method uses the estimate of $\boldsymbol{\theta}_\|^*$, denoted as $\tilde{\boldsymbol{\theta}}_\|$, to perform the "attackability test" by solving LP (4.5) using $\tilde{\boldsymbol{\theta}}_\|$ to obtain an estimated index $\tilde{\epsilon}^*$ and certificate $\tilde{\boldsymbol{\theta}}_\perp$. If $\tilde{\epsilon}^* > 0$, the method asserts the environment is attackable and starts the second stage of attack. From T_1 to T, the adversary perturbs the reward by $\tilde{r} = x^\mathsf{T}(\tilde{\boldsymbol{\theta}}_\| + \tilde{\boldsymbol{\theta}}_\perp)$ (just like the oracle attack but using the estimated $\tilde{\boldsymbol{\theta}}_\|$). When the bandit algorithm pulls the target arm $\tilde{x}$ for the first time in the second stage, the adversary will compensate the reward as shown in line 19. $n(\tilde{x})$ is the number of times target arm is pulled before T_1. The goal is to correct the rewards on $\tilde{x}$ collected in the first stage to follow $\tilde{\boldsymbol{\theta}}$ instead of $\boldsymbol{\theta}_0$. Note that a larger T_1 brings in more observations on $\tilde{x}$ to improve the estimate of $\boldsymbol{\theta}_\|^*$; but it also means a larger attacking cost. The optimal choice of T_1 depends on certain "robustness" property of the bandit algorithm in use. Consequently, it also leads to different amount of attack cost for different algorithms. For example, as we will show below, the attack to Robust Phase Elimination will be more costly than the attack of LinUCB.

Our attackability test might make both false positive and false negative assertions due to the estimation error in $\tilde{\boldsymbol{\theta}}_{\|}$. But as T become large, the estimate is more accurate and the assertion is correct with high probability (see below). We note that an important step in our attack is that the adversary manipulates the rewards for both the targeted arm and other arms in the second stage, as shown in line 21 of Algorithm 8. This is different from the oracle attack where only the rewards of non-target arms are perturbed. This difference turns out to be crucial because it guarantees that the rewards are (almost) always generated by $\tilde{\boldsymbol{\theta}}$, which is the key to the attack's success. Specifically, if the adversary does not perturb the rewards of the target arm $\tilde{x}$ and passes the original rewards generated by $\boldsymbol{\theta}^*$ to the bandit algorithm, these rewards could be viewed as "corrupted" — the corruption comes from the difference between $\tilde{\boldsymbol{\theta}}_{\|}$ and $\boldsymbol{\theta}^*_{\|}$, which may accumulate to a large discrepancy over $T - o(T)$ many rounds' pull of the target arm. This discrepancy may harm our attempts on lowering the bandit algorithm's estimated rewards of non-target arms *due to its correlation with the features of other arms*.[3]

Attack against LinUCB. We now show how LinUCB algorithm can be attacked by Algorithm 8.

Theorem 14. *For large enough T, the attack strategy in Algorithm 8 will correctly assert the attackability with probability at least $1-\delta$. Moreover, when the environment is attackable, with probability at least $1-2\delta$ the attack strategy will fool LinUCB to pull non-target arms at most*

$$O(d(\sqrt{\log(T/\delta)} + \sqrt{T_1}\log(T_1/\delta))\sqrt{T\log(T/\delta)}/\epsilon^*)$$

rounds and the adversary's cost is at most

$$2T_1 + O(T/\sqrt{T_1}) + O(d(\sqrt{\log(T)/\delta} + \sqrt{T_1}\log(T_1/\delta))\sqrt{T\log(T/\delta)}/\epsilon^*),$$

where the last term is due to the manipulation whenever a non-target arm is pulled at the second stage. Specifically, when $T_1 = T^{1/2}$, the strategy gives the minimum attack cost in the order of $\tilde{O}(T^{3/4})$, and the non-target arms are pulled at most $\tilde{O}(T^{3/4})$ rounds.

Proof Sketch. To prove the the assertion is correct with high probability, the idea is that estimated $\tilde{\boldsymbol{\theta}}_{\|}$ is close to the true parameter $\boldsymbol{\theta}^*_{\|}$. We first note that in the first stage, the bandit algorithm will pull the target arm $\tilde{x}$ $T_1 - O(\sqrt{T_1})$ times, since $\tilde{x}$ is the best arm according to $\boldsymbol{\theta}_0$. According to Hoeffding's inequality, the estimation error $\|\tilde{\boldsymbol{\theta}}_{\|} - \boldsymbol{\theta}^*_{\|}\|_2 \le \sqrt{\frac{2\log(2/\delta)}{T_1 - O(\sqrt{T_1})}}$. So with a large enough T_1, the error of $\tilde{x}$'s reward estimation is smaller than ϵ^*. Thus solving LP (4.5) with $\tilde{\boldsymbol{\theta}}_{\|}$ and we could correctly assert attackability with positive estimated index $\tilde{\epsilon}^*$ when the environment is attackable with index ϵ^*.

To proving the success and the cost of the attack, the main challenge lies at analyzing the behavior of LinUCB under the reward discrepancy between the two stages, i.e., corrupted rewards in the first stage. Our proof crucially hinges on the following robustness property of LinUCB.

Lemma 12 (Robustness of ridge regression)**.** *Consider LinUCB with ridge regression for linear stochastic bandits under poisoning attack. For any $t = 1 \dots T$, with probability at least $1-\delta$ we have*

$$\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\|_{A_t} = \alpha_t + S_t/\sqrt{\lambda}$$

where $\boldsymbol{\theta}$ is true bandit parameter, $S_t = \sum_{\tau=1}^{t} |\Delta_\tau|$ is the total corruption until time t, and $\alpha_t = \sqrt{d\log\left(\frac{1+t/\lambda}{\delta}\right)} + \sqrt{\lambda}$. Consequently, the regret of LinUCB can be bounded by $O(d(\sqrt{\log(T/\delta)} + S_t)\sqrt{T\log(T/\delta)})$.

Based on this lemma, we can derive the regret $R(T)$ of LinUCB with $\tilde{\boldsymbol{\theta}}$ as the true parameter. The total corruption is $O(d\sqrt{T_1}\log(T_1/\delta))$ due to the rewards of non-target arms generated by $\boldsymbol{\theta}_0$ in the first stage (the

[3] Previous works [21, 110, 116] do not attack the target arm since in their setting, the reward of target arm would not affect reward estimate of non-target arms. Our problem is harder due to the correlation among arms.

rewards of target arm are compensated in line 19). So the non-target arms are pulled at most $R(T)/\epsilon^*$ rounds. Substitute the total manipulation back and we have the bound.

The attack cost has three sources: 1) attacks in the first stage bounded by $2T_1$; 2) attacks on target the arm in the second stage; and 3) attacks on non-target arms in the second stage. The second term is in $O(T/\sqrt{T_1})$ because the cost per-round $\|\tilde{x}^\intercal(\tilde{\boldsymbol{\theta}}_{\|} - \boldsymbol{\theta}^*_{\|})\|_2$ is in $O(1/\sqrt{T_1})$. The third term has the same order as the number of rounds non-target arms are pulled by LinUCB. By setting $T_1 = T^{1/2}$, the total cost achieves $\tilde{O}(T^{3/4})$. □

Remark 2. *Lemma 12 shows that LinUCB still enjoys sublinear regret for any corruption amount $S = o(\sqrt{T})$. This tolerance of $o(\sqrt{T})$ attack turns out to be the same as the recently proposed robust linear contextual bandit algorithm based on phase elimination in [113] (which we examine next). However, the regret term $S\sqrt{T}$ in LinUCB has a worse dependence on S within the $S = o(\sqrt{T})$ regime compared to the $O(S^2)$ regret dependence of the robust algorithm in [113].*

Attack against Robust Phase Elimination. We now show that Robust Phase Elimination (RobustPhE) can also be attacked by Algorithm 8. Comparing to attacking LinUCB, robustness of this algorithm brings challenge to the first stage as attack cost is more sensitive to the length of this stage.

Corollary 3. *For any large enough T, the attack strategy in Algorithm 8 will correctly assert the attackability with high probability. Moreover, when the environment is attackable, with probability at least $1 - \delta$ the attack strategy will fool RobustPhE to pull non-target arms at most*

$$O((d\sqrt{T}\log(T/\delta) + T_1^2)/\epsilon^*)$$

rounds and the adversary spends cost at most

$$2T_1 + O(T/\sqrt{T_1}) + O((d\sqrt{T}\log(T/\delta) + T_1^2)/\epsilon^*)$$

where the last term is due to the manipulation whenever a non-target arm is pulled at the second stage. Specifically, $T_1 = T^{2/5}$ gives the minimum attack cost order $\tilde{O}(T^{4/5})$ and the non-target arms are pulled at most $\tilde{O}(T^{4/5})$ rounds.

Robust Phase Elimination has an additional regret term $O(S^2)$ for total corruption S (assuming S is unknown to the bandit algorithm). If we view the second stage attack model $\tilde{\boldsymbol{\theta}}$ as the underlying environment bandit model, then rewards generated by $\boldsymbol{\theta}_0$ in the first stage are corrupted rewards. The T_1 amount of rewards from the first stage mean T_1 corruption, which leads to the additional T_1^2 term in the cost and the number of non-target arm pulls compared with Theorem 14. Hence, the adversary can only run fewer iterations in the first stage but spends more budget there. On the other hand, this also favors the design of attack such that line 18-19 in Algorithm 8 is not necessary: the corruption in the first stage can be handled by the robustness of bandit algorithm. Our success of attacking RobustPhE does not violate the robustness claim in the original paper [113]: RobustPhE could tolerate $O(\sqrt{T})$ corruption and our attack cost is $\tilde{O}(T^{4/5})$.

4.2.4 Experiments

We use simulation-based experiments to validate the effectiveness and cost-efficiency of our proposed attack methods. In our simulations, we generate a size-k arm pool $\mathcal{A}$, in which each arm a is associated with a context vector $x_a \in \mathbb{R}^d$. Each dimension of x_a is drawn from a set of zero-mean Gaussian distributions with variances sampled from a uniform distribution $U(0, 1)$. Each v_a is then normalized to $\|x_a\|_2 = 1$. The bandit model parameter $\boldsymbol{\theta}^*$ is sampled from $N(0, 1)$ and normalized to $\|\boldsymbol{\theta}^*\|_2 = 1$. We set d to 10, the standard derivation σ of Gaussian noise η_t to 0.1, and the arm pool size k to 30 in our simulations. We run the experiment for $T = 10,000$ iterations. We will re-sample the environment $\langle \mathcal{A}, \boldsymbol{\theta}^*, \tilde{x}\rangle$ until it is attackable, following Theorem 13.

We compare the two proposed attack methods, oracle null space attack and two-stage null space attack, against LinUCB [12] and Robust Phase Elimination (RobustPhE) [113]. We report average results of 10 runs where in each run we sample a random attackable environment. Both oracle attack and two-stage attack

can effectively fool the two bandit algorithms to pull the target arm linear times and we report this result in appendix. Figure 4.4 shows the total cost of the attack. We observe that both attack methods are cost-efficient with sublinear total cost, while two-stage attack requires more attack budget. Specifically, we notice that the adversary spends almost linear budget in the first stage. This is because in the first stage the adversary attacks according to parameter $\boldsymbol{\theta}_0$ which leads to a almost constant cost at every round. In the second stage, the cost is much smaller: the adversary only spends $O(1/\sqrt{T_1})$ cost when pulling the target arm. This also corresponds to our theoretical analysis that total cost of two-stage attack is $O(T^{3/4})$ against LinUCB and $O(T^{4/5})$ against RobustPhE. To attack the same bandit algorithm, the total cost of two-stage attack is larger than oracle attack. The key reason is that when pulling target arm, the oracle attack does not perturb the reward. We see that cost does not increase in oracle attack against LinUCB in the later stage, but the curve of two-stage attack against LinUCB keeps increase over time. We also notice that for the same attack method, attacking RobustPhE requires more budget and the target arm pull is also smaller comparing with attacking LinUCB, due to the robustness of the algorithm.

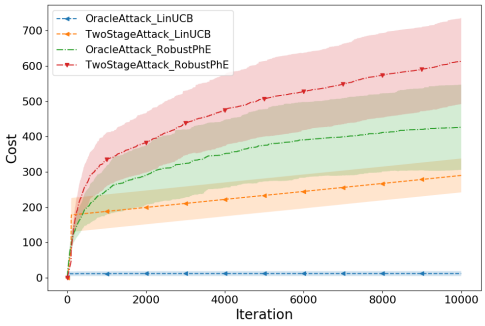


Figure 4.4: Total cost of the attacks.

Chapter 5

Conclusion & Future Work

This thesis aims to understand the role of structural information in interactive online learning problems. Our research solves several key challenges in the learning by exploration paradigm including the problem of huge exploration space, missing information and privacy and security concerns. We provided a deep and thorough understanding the benefit of leveraging structural information as an advantage and extend the application of bandit learning algorithms to more complex practical scenarios. By combining the proposed techniques, an advanced intelligent system can leverage the right information to serve and interact with humans in an efficient and trustful manner. Rigorous theoretical analysis and extensive empirical evaluation validated the approaches' applicability in various contexts and applications.

In Chapter 2, we introduced using explicit structural information to reduce the huge exploration space and achieve sample efficient exploration. We developed collaborative linear bandit algorithm CoLin for recommender system, which utilized social network for information sharing among the neighboring users during online update and reduced sample complexity. We also developed Document Space Projection for dueling bandit based online learning to rank, which identified the low-rank gradient space of the ranking problem and explored more efficiently in the projected space. In Chapter 3, we considered the environment with implicit (unobservable) structure. We presented factorization bandits to recover latent factors in a low-rank environment. We also identified the problem of information gap in a two-party game between an interactive system and the users. We showed that when the context information on the user side is unknown to the system, the system can incentivize users to explore even with such information disadvantage as long as the users' contexts follow the structure of linear transformation to the contexts on the system side. In Chapter 4, we discussed privacy and security aspects of interactive online learning. We developed a framework to equipment collaborative linear bandit algorithms such as CoLin with global and local differential privacy guarantee, and showed the structural information helped to improve the privacy-utility trade-off. We also discussed the potential vulnerability of the linear bandit algorithms to data poisoning attacks. We showed that the attackability is determined by the geometric structure of the context features and presented attacking strategies that can manipulate the behavior of a linear bandit algorithm.

Our research on learning by exploration with information advantage for interactive intelligent systems sheds lights on important yet challenging future directions.

Multi-agent interactive learning with information gap

Real-world applications often involve interactions with multiple agents, while agents act in their own strategic way. For example, in cyber-physical systems and self-driving cars, users/drivers and hundreds of sensors and devices act as interactive agents that communicate and collaborate with others; in ads auctions, different advertisers who compete with each other to purchase ads with a minimum cost. Each agent could generate and receive a huge amount of information yet ignore or cannot access even larger amount of information. The *information gap* among them, i.e., the known and unknown about other agents and the uncertainty in the environment, plays an important role for an agent to realize its optimal action. Our research on collaborative

bandit learning offered a preliminary understanding on a collaborative multi-agent bandit environment, and incentivizing exploration under information gap research studied the price of information gap in a two-party game. We believe it is important and promising to pursue a fundamental study on *the role of information in multi-agent interactive learning*. Each agent's action is a result of other agents' actions; and in turn they reveal information that affects other agents' beliefs and decisions. The agents thus can take advantage of the information gap for exploring the environment and finding the optimal action, e.g., take calibrated actions to influence and persuade other agents. Understanding the agents' behavior requires domain knowledge and is an interdisciplinary research problem. With the collaboration with experts in different applications, we plan to develop interactive online learning algorithms that leverage information from other agents in a collaborative environment or exploit the information gap in a competitive environment to efficiently explore the problem space.

Ethical and security considerations

While important and exciting to design advanced interactive systems for humans by exploiting the information advantage, it is equally important, if not more, to develop *trustworthy* systems and use them for *social good*. As intelligent learning systems are already pervasive in our daily lives, ethical and security considerations of intelligent system are now more than a public opinion and become a legal requirement, e.g., European Union's GDPR[1]. We believe another promising direction is to develop interactive online learning systems the following ethical and security aspects.

Privacy. Privacy is a critical concern for online learning algorithms that directly interact with humans. It is important to consider the potential privacy breaches by an interactive learning system when personal information is involved, e.g., personal preference in recommendation and ads display. Beyond our previous studies in private recommender system, it is also important to investigate the interactive online learning solutions with privacy guarantee for more challenging applications such as ranking systems, cyber-physical systems and health care with problem-specific structure in consideration. **Fairness.** Fairness is another important ethical constraint for online decision making. Understanding how to use the right information to make a fair decision is critical: the decisions would have significant consequences on people's lives in applications like hiring and education. It is interesting to consider the trade-off between utility and fairness guarantee under different fairness definitions. **Robustness and Security.** In many real-world scenarios, the learning environment is not always benign. Various challenges exist in real-world situations such as feedback bias and adversarial attacks. Our research on data poisoning attacks on linear bandits already showed the potential vulnerability of an interactive online learning system. Thus it is necessary to consider the online learning process in an adversarial setting with robustness guarantee. **Safety.** The performance of an interactive online learning algorithm, such as multi-armed bandit and reinforcement learning, can vary drastically during online learning. Due to safety concerns, intelligent systems for high-stake applications like self-driving cars and healthcare are expected to have high-quality decision making with guarantees. Systems for high-stake applications require new algorithms whose performance variance can be quantified and guaranteed during online learning.

[1] https://gdpr-info.eu/

Bibliography

[1] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538. ACM, 2016.

[2] Huazheng Wang, Qingyun Wu, and Hongning Wang. Factorization bandits for interactive recommendation. In *AAAI*, pages 2695–2702, 2017.

[3] Huazheng Wang, Ramsey Langley, Sonwoo Kim, Eric McCord-Snook, and Hongning Wang. Efficient exploration of gradient space for online learning to rank. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 145–154. ACM, 2018.

[4] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. Variance reduction in gradient exploration for online learning to rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 835–844, 2019.

[5] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 636–646, 2019.

[6] John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 148–177, 1979.

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

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

[9] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

[10] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[11] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. *In Pro. NIPS*, 2013.

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

[13] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, pages 586–594, 2010.

[14] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *Neural Information Processing*, pages 324–331. 2012.

[15] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *NIPS 2011*, pages 2249–2257, 2011.

[16] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. Exploitation and exploration in a performance based contextual advertising system. In *Proceedings of 16th SIGKDD*, pages 27–36. ACM, 2010.

[17] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1633–1642. ACM, 2016.

[18] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. " you might also like:" privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy*, pages 231–246. IEEE, 2011.

[19] Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *ICDM 2010*, pages 474–482. IEEE, 2010.

[20] Huazheng Wang, Qian Zhao, Qingyun Wu, Shubham Chopra, Abhinav Khaitan, and Hongning Wang. Global and local differential privacy for collaborative bandits. In *Fourteenth ACM Conference on Recommender Systems*, pages 150–159, 2020.

[21] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3640–3649, 2018.

[22] Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050, 2019.

[23] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Multi-armed bandits in the presence of side observations in social networks. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7309–7314. IEEE, 2013.

[24] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. *SIGMETRICS Perform. Eval. Rev.*, 42(1):289–300, June 2014.

[25] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *ICML'14*, pages 757–765, 2014.

[26] S. Kar, H.V. Poor, and Shuguang Cui. Bandit problems in networks: Asymptotically efficient distributed allocation rules. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 1771–1778, Dec 2011.

[27] Kareem Amin, Michael Kearns, and Umar Syed. Graphical models for bandit problems. *Proceedings of UAI 2011*, 2011.

[28] Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

[29] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, pages 1297–1305, 2015.

[30] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd CIKM*, pages 1411–1420. ACM, 2013.

[31] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of 4th WSDM*, pages 297–306. ACM, 2011.

[32] Artem Grotov and Maarten de Rijke. Online learning to rank for information retrieval: Sigir 2016 tutorial. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1215–1218. ACM, 2016.

[33] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[34] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM CIKM*, pages 43–52. ACM, 2008.

[35] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of 26th ICML*, pages 1201–1208. ACM, 2009.

[36] Yisong Yue, Yue Gao, Oliver Chapelle, Ya Zhang, and Thorsten Joachims. Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 507–514. ACM, 2010.

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

[38] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, volume 51, pages 4–11. Acm, 2017.

[39] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for ir. In *Proceedings of the sixth ACM international conference on WSDM*, pages 183–192. ACM, 2013.

[40] Harrie Oosterhuis and Maarten de Rijke. Balancing speed and quality in online learning to rank for information retrieval. In *Proceedings of the 2017 ACM CIKM*, pages 277–286. ACM, 2017.

[41] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94. ACM, 2008.

[42] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[43] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. Multileave gradient descent for fast online learning to rank. In *Proceedings of the Ninth ACM International Conference on WSDM*, pages 457–466. ACM, 2016.

[44] Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. Multileaved comparisons for fast online evaluation. In *Proceedings of the 23rd ACM CIKM*, pages 71–80. ACM, 2014.

[45] Tong Zhao and Irwin King. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1643–1652. ACM, 2016.

[46] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776, 2015.

[47] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Dcm bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning*, pages 1215–1224, 2016.

[48] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *International Conference on Machine Learning*, pages 4199–4208, 2017.

[49] Harrie Oosterhuis and Maarten de Rijke. Differentiable unbiased online learning to rank. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18*, 2018.

[50] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 249–258. ACM, 2011.

[51] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.

[52] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, volume 310, 2007.

[53] Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets, 2013.

[54] Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24, 2011.

[55] Ellen M Voorhees, Donna K Harman, et al. *TREC: Experiment and evaluation in information retrieval*, volume 1. MIT press Cambridge, 2005.

[56] Anne Schuth, Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Lerot: An online learning to rank framework. In *Proceedings of the 2013 workshop on Living labs for information retrieval evaluation*, pages 23–26. ACM, 2013.

[57] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[58] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[59] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[60] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[61] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[62] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[63] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD*, pages 19–28. ACM, 2009.

[64] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[65] Atsuyoshi Nakamura. A ucb-like strategy of collaborative filtering. In *ACML*, 2014.

[66] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548. ACM, 2016.

[67] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[68] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[69] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320. 2011.

[70] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 635–644, New York, NY, USA, 2011. ACM.

[71] André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2):639–652, 2012.

[72] Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the "wisdom of the crowd". *Journal of Political Economy*, 122(5):988–1012, 2014.

[73] Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 5–22. ACM, 2014.

[74] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, volume 68, pages 565–582. ACM, INFORMS, 2015.

[75] Christoph Hirnschall, Adish Singla, Sebastian Tschiatschek, and Andreas Krause. Learning user preferences to incentivize exploration in the sharing economy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[76] Siwei Wang and Longbo Huang. Multi-armed bandits with compensation. In *NeurIPS*, 2018.

[77] Ingmar Weber and Carlos Castillo. The demographics of web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 523–530, 2010.

[78] Bin Bi, Milad Shokouhi, Michal Kosinski, and Thore Graepel. Inferring the demographics of search users: Social data meets search queries. In *Proceedings of the 22nd international conference on World Wide Web*, pages 131–140, 2013.

[79] Aleksandrs Slivkins. Incentivizing exploration via information asymmetry. *XRDS: Crossroads, The ACM Magazine for Students*, 24(1):38–41, 2017.

[80] Nicole Immorlica, Jieming Mao, Aleksandrs Slivkins, and Zhiwei Steven Wu. Incentivizing exploration with selective data disclosure. *arXiv preprint arXiv:1811.06026*, 2018.

[81] Mark Sellke and Aleksandrs Slivkins. Sample complexity of incentivized exploration. *arXiv preprint arXiv:2002.00558*, 2020.

[82] Bangrui Chen, Peter Frazier, and David Kempe. Incentivizing exploration by heterogeneous users. In *Conference On Learning Theory*, pages 798–818. PMLR, 2018.

[83] Priyank Agrawal and Theja Tulabandhula. Incentivising exploration and recommendations for contextual bandits with payments. In *Multi-Agent Systems and Agreement Technologies*, pages 159–170. Springer, 2020.

[84] Zhiyuan Liu, Huazheng Wang, Fan Shen, Kai Liu, and Lijun Chen. Incentivized exploration for multi-armed bandits under reward drift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4981–4988, 2020.

[85] Sampath Kannan, Michael Kearns, Jamie Morgenstern, Mallesh Pai, Aaron Roth, Rakesh Vohra, and Zhiwei Steven Wu. Fairness incentives for myopic agents. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 369–386, 2017.

[86] Sahin Lale, Kamyar Azizzadenesheli, Anima Anandkumar, and Babak Hassibi. Stochastic linear bandits with hidden low rank structure. *arXiv preprint arXiv:1901.09490*, 2019.

[87] Jiaqi Yang, Wei Hu, Jason D. Lee, and Simon Shaolei Du. Impact of representation learning in linear bandits. In *International Conference on Learning Representations*, 2021.

[88] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *AISTATS'11*, pages 208–214, 2011.

[89] Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Artificial Intelligence and Statistics*, pages 728–737. PMLR, 2017.

[90] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pages 2733–2741, 2013.

[91] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *ICML 2017*, pages 32–40. JMLR. org, 2017.

[92] Aristide Charles Yedia Tossou and Christos Dimitrakakis. Achieving privacy in the adversarial multi-armed bandit. In *AAAI 2017*, 2017.

[93] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 4296–4306, 2018.

[94] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[95] Seth Neel and Aaron Roth. Mitigating bias in adaptive data gathering via differential privacy. *arXiv preprint arXiv:1806.02329*, 2018.

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

[97] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[98] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[99] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724. ACM, 2010.

[100] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):26, 2011.

[101] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD*, pages 627–636. ACM, 2009.

[102] Tianqing Zhu, Gang Li, Yongli Ren, Wanlei Zhou, and Ping Xiong. Differential privacy for neighborhood-based collaborative filtering. In *ASONAM 2013*, pages 752–759. ACM, 2013.

[103] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *RecSys 2015*, pages 171–178. ACM, 2015.

[104] Prateek Jain, Om Dipakbhai Thakkar, and Abhradeep Thakurta. Differentially private matrix completion revisited. In *ICML*, pages 2215–2224, 2018.

[105] Jacob Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. *arXiv preprint arXiv:1711.10019*, 2017.

[106] Nikita Mishra and Abhradeep Thakurta. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 592–601. AUAI Press, 2015.

[107] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *SIGSAC 2014*, pages 1054–1067. ACM, 2014.

[108] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.

[109] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[110] Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Jean Tarbouriech, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits, 2020.

[111] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474, 2012.

[112] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[113] Ilija Bogunovic, Arpan Losalka, Andreas Krause, and Jonathan Scarlett. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR, 2021.

[114] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. *arXiv preprint arXiv:2003.12613*, 2020.

[115] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 7974–7984. PMLR, 2020.

[116] Yang Liu and Chien-Ju Ho. Incentivizing high quality user contributions: new arm generation in bandit learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[117] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, page 87–94, New York, NY, USA, 2008. ACM, Association for Computing Machinery.

[118] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: Dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, page 113–120, New York, NY, USA, 2014. Association for Computing Machinery.

[119] Daniel Weitekamp, Erik Harpstead, and Ken R. Koedinger. An interaction design for machine teaching to develop ai tutors. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–11, New York, NY, USA, 2020. Association for Computing Machinery.

[120] Charu C. Aggarwal. *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[121] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, and M. Wang. Modeling the evolution of users #8217; preferences and social links in social networking services. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1240–1253, June 2017.

[122] 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.

[123] Naoki Abe and Atsuyoshi Nakamura. Learning to optimally schedule internet banner advertisements. In *ICML*, volume 99, pages 12–21, 1999.

[124] Naoki Abe, Alan W Biermann, and Philip M Long. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4):263–293, 2003.

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

[126] Alex Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT08'*, pages 343–354, July 2008.

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

[128] Sinan Aral, Lev Muchnik, and Arun Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[129] Benjamin Nye. Intelligent tutoring systems by and for the developing world: A review of trends and approaches for educational technology in a global context. *International Journal of Artificial Intelligence in Education*, 25, 06 2015.

[130] Shuo Tian, Jehane Michael Le Grange, Peng Wang, Wei Huang, and Zhewei Ye. Smart healthcare: making medical care more intelligent. *Global Health Journal*, 3, 10 2019.

[131] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[132] Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 116–120, 2016.

[133] Baruch Awerbuch and Robert Kleinberg. Competitive collaborative learning. *J. Comput. Syst. Sci.*, 74(8):1271–1288, December 2008.

[134] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[135] Lilian Besson and Emilie Kaufmann. The generalized likelihood ratio test meets klucb: an improved algorithm for piece-wise non-stationary bandits. *arXiv preprint arXiv:1902.01575*, 2019.

[136] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Technical Report MSR-TR-98-12, Microsoft Research, May 1998.

[137] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors. *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.

[138] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[139] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[140] R Stephen Cantrell, Peter M Burrows, and Quang H Vuong. Interpretation and use of generalized chow tests. *International Economic Review*, pages 725–741, 1991.

[141] Yang Cao, Wen Zheng, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure for piecewise-stationary bandit: a change-point detection approach. *AISTATS,(Okinawa, Japan)*, 2019.

[142] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[143] Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *J. Mach. Learn. Res.*, 11:2901–2934, December 2010.

[144] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

[145] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[146] Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. *arXiv preprint arXiv:1902.00980*, 2019.

[147] Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, pages 591–605, 1960.

[148] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 815–824, New York, NY, USA, 2016. ACM.

[149] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.

[150] Robert B Cialdini and Melanie R Trost. Social influence: Social norms, conformity and compliance. 1998.

[151] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658. ACM, 2018.

[152] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

[153] Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: stochastic and adversarial bandits. *CoRR*, abs/1202.4473, 2012.

[154] Haipeng Luo, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. *arXiv preprint arXiv:1708.01799*, 2017.

[155] Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. *CoRR*, abs/1704.06880, 2017.

[156] Julien Delporte, Alexandros Karatzoglou, Tomasz Matuszczyk, and Stéphane Canu. Socially enabled preference learning from implicit feedback data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–160. Springer, 2013.

[157] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[158] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare Conference*, pages 67–82, 2018.

[159] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[160] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, pages 817–824. 2008.

[161] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the 10th ACM SIGKDD*, pages 109–117. ACM, 2004.

[162] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005.

[163] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. In *arXiv preprint arXiv:0805.3415 (2008)*.

[164] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, ALT'11, pages 174–188, Berlin, Heidelberg, 2011. Springer-Verlag.

[165] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1253–1262. JMLR. org, 2017.

[166] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the Second ACM International Conference on WSDM*, pages 124–131. ACM, 2009.

[167] Li Han, David Kempe, and Ruixin Qiang. Incentivizing exploration with heterogeneous value of money. In *International Conference on Web and Internet Economics*, pages 370–383. Springer, 2015.

[168] Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 4268–4274. AAAI Press, 2015.

[169] Cedric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michele Sebag. Multi-armed Bandit, Dynamic Environments and Meta-Bandits. November 2006.

[170] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1):63–90, 2013.

[171] Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 557–566. ACM, 2013.

[172] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

[173] Ali Alkhatlan and Jugal Kalita. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *International Journal of Computer Applications*, 181(43):1–20, Mar 2019.

[174] Tommi S Jaakkola and Michael I Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, 2000.

[175] Shweta Jain, Balakrishnan Narayanaswamy, and Y. Narahari. A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 721–727. AAAI Press, 2014.

[176] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD*, pages 133–142. ACM, 2002.

[177] Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. *Adversarial Machine Learning*. Cambridge University Press, 2018.

[178] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

[179] Michael Kearns, Mallesh Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 403–410. ACM, 2014.

[180] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. In *Acm Sigir Forum*, volume 37, pages 18–28. ACM, 2003.

[181] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

[182] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.

[183] Wataru Kumagai. Regret analysis for continuous dueling bandit. In *Advances in Neural Information Processing Systems*, pages 1489–1498, 2017.

[184] Yanyan Lan, Tie-Yan Liu, Tao Qin, Zhiming Ma, and Hang Li. Query-level stability and generalization in learning to rank. In *Proceedings of the 25th international conference on Machine learning*, pages 512–519. ACM, 2008.

[185] Yanyan Lan, Tie-Yan Liu, Zhiming Ma, and Hang Li. Generalization analysis of listwise learning-to-rank algorithms. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 577–584. ACM, 2009.

[186] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. 2020.

[187] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR*, SIGIR '16, pages 539–548, New York, NY, USA, 2016. ACM, ACM.

[188] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *International Conference on Machine Learning*, pages 1245–1253, 2016.

[189] Jianqiang Li, Ji-Jiang Yang, Yu Zhao, Bo Liu, Mengchu Zhou, Jing Bi, and Qing Wang. Enforcing differential privacy for shared collaborative filtering. *IEEE Access*, 5:35–49, 2016.

[190] 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.

[191] Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[192] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[193] Fang Liu, Joohyun Lee, and Ness Shroff. A change-detection based framework for piecewise-stationary multi-armed bandit problem. AAAI'18, 2018.

[194] Zhiyuan Liu, Huazheng Wang, Bo Waggoner, Lijun Chen, et al. A smoothed analysis of online lasso for the sparse linear contextual bandit problem. *Workshop on Real World Experiment Design and Active Learning at ICML 2020*, 2020.

[195] T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[196] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122. ACM, 2018.

[197] Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, pages 186–204. Springer, 2018.

[198] Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pages 14543–14553, 2019.

[199] Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144, 2014.

[200] Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[201] Benjamin Marlin and Richard S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 73–, New York, NY, USA, 2004. ACM.

[202] Kentaro Minami, HItomi Arai, Issei Sato, and Hiroshi Nakagawa. Differential privacy without sensitivity. In *Advances in Neural Information Processing Systems*, pages 956–964, 2016.

[203] Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. 1985.

[204] Eshcar Hillel, Zohar S Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 854–862. Curran Associates, Inc., 2013.

[205] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 199–207. Curran Associates, Inc., 2014.

[206] Zohar S Karnin and Oren Anava. Multi-armed bandits: Competing with optimal sequences. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *NIPS'16*, pages 199–207. Curran Associates, Inc., 2016.

[207] Yi Qi, Qingyun Wu, Hongning Wang, Jie Tang, and Maosong Sun. Bandit learning with implicit feedback. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7276–7286. Curran Associates, Inc., 2018.

[208] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

[209] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.

[210] Umut Ozertem, Olivier Chapelle, Pinar Donmez, and Emre Velipasaoglu. Learning to suggest: a machine learning framework for ranking query suggestions. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 25–34. ACM, 2012.

[211] Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 116–120, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[212] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In Doina Precup and Yee Whye Teh, editors,

ICML'17, volume 70 of *Proceedings of Machine Learning Research*, pages 1253–1262, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[213] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of 25th ICML*, pages 784–791. ACM, 2008.

[214] Anshuka Rangi, Long Tran-Thanh, Haifeng Xu, and Massimo Franceschetti. Secure-ucb: Saving stochastic bandits from poisoning attacks via limited data verification, 2021.

[215] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.

[216] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM, ACM.

[217] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 09 1952.

[218] Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026, 2019.

[219] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[220] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[221] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. Citeseer, 2011.

[222] Mark Sanderson et al. Test collection based evaluation of information retrieval systems. *Foundations and Trends® in Information Retrieval*, 4(4):247–375, 2010.

[223] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[224] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, pages 158–166, New York, NY, USA, 1999. ACM.

[225] J Ben Schafer, Joseph A Konstan, and John Riedl. E-commerce recommendation applications. In *Applications of Data Mining to Electronic Commerce*, pages 115–153. Springer, 2001.

[226] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of 25th SIGIR*, pages 253–260. ACM, 2002.

[227] Muzafer Sherif. The psychology of social norms. 1936.

[228] Milad Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 103–112. ACM, 2013.

[229] David Siegmund. *Sequential analysis: tests and confidence intervals*. Springer Science & Business Media, 2013.

[230] Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(Feb):399–436, 2013.

[231] Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

[232] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 53–60, New York, NY, USA, 2009. ACM.

[233] Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In *ICML*, volume 3, pages 720–727, 2003.

[234] Stephen M. Stigler. The asymptotic distribution of the trimmed mean. *Ann. Statist.*, 1(3):472–477, 05 1973.

[235] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[236] 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.

[237] Aristide CY Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *AAAI 2016*, 2016.

[238] S.S. Villar, J. Bowden, and J. Wason. Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. In *Statist. Sci., 30 (2) (2015)*, pages 199–215, 2015.

[239] Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. Exploring compact reinforcement-learning representations with linear regression. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 591–598. AUAI Press, 2009.

[240] Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. Bandit problems with side observations. *Automatic Control, IEEE Transactions on*, 50(3):338–355, 2005.

[241] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen W White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the 37th international ACM SIGIR conference*, pages 123–132. ACM, 2014.

[242] Yizhen Wang and Kamalika Chaudhuri. Data poisoning attacks against online learning, 2018.

[243] Paul M Weichsel. The kronecker product of graphs. *Proceedings of the American Mathematical Society*, 13(1):47–52, 1962.

[244] AL Wilson. When is the chow test ump? *The American Statistician*, 32(2):66–68, 1978.

[245] Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *Proceedings of the 41th International ACM SIGIR*, pages 495–504. ACM, 2018.

[246] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, pages 2080–2090, 2019.

[247] Qingyun Wu, Chi Wang, and Silu Huang. Cost effective optimization for cost-related hyperparameters, 2020.

[248] Qingyun Wu, Huazheng Wang, and Hongning Wang. Learning by exploration: New challenges in real-world environments. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3575–3576, 2020.

[249] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhaohui Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 295–304. ACM, 2011.

[250] Jia Yuan Yu and Shie Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1177–1184, New York, NY, USA, 2009. ACM, ACM.

[251] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, pages 2483–2491, 2011.

[252] Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401, 2016.

[253] Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attack, 2019.

[254] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *International Conference on Algorithmic Applications in Management*, pages 337–348. Springer, 2008.

[255] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

[256] Alon Zweig and Gal Chechik. Group online adaptive learning. *Machine Learning*, 106(9):1747–1770, Oct 2017.

[257] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[258] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

[259] Zhe Feng, David C Parkes, and Haifeng Xu. The intrinsic robustness of stochastic bandits to strategic manipulation. *arXiv preprint arXiv:1906.01528*, 2019.

[260] Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5):30, 2017.

[261] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

[262] Milton Abramowitz. Handbook of mathematical functions with formulas. *Graphs, and Mathematical Tables*, 1965.

[263] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[264] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, pages 1–9. IEEE, 2010.

[265] Ismail Razak, Nazief Nirwanto, and Boge Triatmanto. The impact of product quality and price on customer satisfaction with the mediator of customer value. *Journal of Marketing and Consumer Research*, 30(1):59–68, 2016.

[266] Anne Martensen, Lars Gronholdt, and Kai Kristensen. The drivers of customer satisfaction and loyalty: cross-industry findings from denmark. *Total Quality Management*, 11(4-6):544–553, 2000.

[267] Zahra Ehsani and Mohammad Hossein Ehsani. Effect of quality and price on customer satisfaction and commitment in iran auto industry. *International Journal of Service Science, Management and Engineering*, 1(5):52, 2015.

[268] Gwo-Guang Lee and Hsiu-Fen Lin. Customer perceptions of e-service quality in online shopping. *International Journal of Retail & Distribution Management*, 33(2):161–176, 2005.

[269] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer, 2005.

[270] Nicole Immorlica, Jieming Mao, Aleksandrs Slivkins, and Zhiwei Steven Wu. Bayesian exploration with heterogeneous agents. In *The World Wide Web Conference*, pages 751–761. ACM, 2019.

[271] Kostas Bimpikis and Yiangos Papanastasiou. Inducing exploration in service platforms. In *Sharing Economy*, pages 193–216. Springer, 2019.

[272] Amin Rakhsha, Xuezhou Zhang, Xiaojin Zhu, and Adish Singla. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *arXiv preprint arXiv:2102.08492*, 2021.

[273] Yiding Chen and Xiaojin Zhu. Optimal attack against autoregressive models by manipulating the environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3545–3552, 2020.

[274] Tiancheng Yu and Suvrit Sra. Efficient policy learning for non-stationary mdps under adversarial manipulation. *arXiv preprint arXiv:1907.09350*, 2019.

[275] Shiliang Zuo. Near optimal adversarial attack on ucb bandits. *arXiv preprint arXiv:2008.09312*, 2020.

[276] Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John Lui, Wing Shing Wong, et al. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. *Advances in Neural Information Processing Systems*, 33, 2020.

[277] Anupam Gupta, Tomer Koren, and Kunal Talwar. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 1562–1578. PMLR, 2019.

[278] Abbasi yadkori et al. Improved algorithms for linear stochastic bandits. In *NIPS*. 2011.

[279] Botao Hao, Tor Lattimore, and Csaba Szepesvari. Adaptive exploration in linear contextual bandit. In *International Conference on Artificial Intelligence and Statistics*, pages 3536–3545. PMLR, 2020.

[280] Andrea Tirinzoni, Matteo Pirotta, Marcello Restelli, and Alessandro Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. *arXiv preprint arXiv:2010.12247*, 2020.

Appendix

A Proofs of the Theorems of Section 2.1

Proof of Lemma 1. Consider the objective function of ridge regression defined in Eq (2.3). By taking the gradient of $L(\mathbf{\Theta})$ with respect to $\mathbf{\Theta}$ and applying our model assumption specified in Eq (2.2), we have,

$$\mathbf{A}_t(\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*) = \sum_{t'=1}^{t} vec(\mathring{\mathbf{X}}_{a_{t'},u'_t}\mathbf{W}^\mathsf{T})\epsilon_{t'} - \lambda\boldsymbol{\vartheta}^*$$

in which $\epsilon_{t'}$ is the Gaussian noise at time t' in reward generation.

Define $\mathbf{S}_t = \sum_{t'=1}^{t} vec(\mathring{\mathbf{X}}_{a_{t'},u'_t}\mathbf{W}^\mathsf{T})\epsilon_{t'}$, we have,

$$\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^* = \mathbf{A}_t^{-1}(\mathbf{S}_t - \lambda\boldsymbol{\vartheta}^*)$$

Because $\mathbf{S}_t$ is a martingale, according to Theorem 1 and 2 in [69],

$$\|\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{2\ln(\frac{det(\mathbf{A}_t)^{1/2}det(\lambda\mathbf{I})^{-1}}{\delta})} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| \tag{1}$$

Since $\|\mathbf{x}_{a_{t,i}}\| \leq 1$, $trace(\mathbf{A}_t) \leq \lambda dN + \sum_{t'=1}^{t}\sum_{j=1}^{N} w^2_{u'_t j}$, we have $det(\mathbf{A}_t) \leq (\frac{trace(\mathbf{A}_t)}{dN})^{dN} \leq (\lambda + \frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w^2_{u'_t j}}{dN})^{dN}$. Similarly, we have $det(\lambda\mathbf{I}_{dN}) \leq \lambda^{dN}$. Putting all these into Eq (1), we have,

$$\|\hat{\boldsymbol{\vartheta}}_t - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\ln(1 + \frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w^2_{u'_t j}}{\lambda dN}) - 2\ln(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\|$$

□

Proof of Theorem 1:

Proof of Theorem 1. According to the definition of regret, the regret of CoLin at time t can be written as,

$$\begin{aligned}
R_t &= r_{a_t^*,u_t} - r_{a_t,u_t} \\
&= vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\boldsymbol{\vartheta}^* - vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\boldsymbol{\vartheta}^* \\
&\leq vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\hat{\boldsymbol{\vartheta}}_{t-1} + \alpha_t\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})\|_{\mathbf{A}_{t-1}^{-1}} - vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})^\mathsf{T}\boldsymbol{\vartheta}^* \\
&\leq \|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})\|_{\mathbf{A}_{t-1}^{-1}}\|\hat{\boldsymbol{\vartheta}}_{t-1} - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_{t-1}} + \alpha_t\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})\|_{\mathbf{A}_{t-1}^{-1}} \\
&\leq 2\alpha_t\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\mathsf{T})\|_{\mathbf{A}_{t-1}^{-1}}
\end{aligned}$$

where the first inequality is based on the following two inequalities,

(1) Based on CoLin's arm selection strategy, if arm $\mathbf{X}_t$ is chosen at time t, it must satisfy,

$$\begin{aligned} &vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\intercal)^\intercal\hat{\boldsymbol{\vartheta}}_{t-1} + \alpha_t\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} \\ &\geq vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\hat{\boldsymbol{\vartheta}}_{t-1} + \alpha_t\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} \end{aligned}$$

(2) Based on Cauchy-Schwarz inequality, we have,

$$\begin{aligned} &vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\hat{\boldsymbol{\vartheta}}_{t-1} + \alpha_t\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} - vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\boldsymbol{\vartheta}^* \\ &\geq -\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\|_{\mathbf{A}_{t-1}^{-1}}\|\hat{\boldsymbol{\vartheta}}_{t-1} - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_{t-1}} + \alpha_t\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} \\ &\geq -\alpha_{t-1}\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} + \alpha_t\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} \\ &\geq 0 \end{aligned}$$

and therefore, we have

$$vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\hat{\boldsymbol{\vartheta}}_{t-1} + \alpha_t\|vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)\|_{\mathbf{A}_{t-1}^{-1}} \geq vec(\mathring{\mathbf{X}}^*_{u_t}\mathbf{W}^\intercal)^\intercal\boldsymbol{\vartheta}^*$$

And according to Lemma 11 in [69], we have,

$$\ln(\frac{det(\mathbf{A}_T)}{det(\lambda\mathbf{I})}) \leq \sum_{t=1}^{T}\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\intercal)\|^2_{\mathbf{A}_{t-1}^{-1}} \leq 2\ln(\frac{det(\mathbf{A}_T)}{det(\lambda\mathbf{I})})$$

Thus the accumulated regret at time T in CoLin can be bounded by,

$$\begin{aligned} \mathbf{R}(T) &\leq \sqrt{T\sum_{t=1}^{T}R_t^2} \leq \sqrt{T4\alpha_T^2\sum_{t=1}^{T}\|vec(\mathring{\mathbf{X}}_{u_t}\mathbf{W}^\intercal)\|^2_{\mathbf{A}_{t-1}^{-1}}} \\ &= \sqrt{T8\alpha_T^2\ln\Big(\frac{det(\mathbf{A}_T)}{det(\lambda\mathbf{I})}\Big)} \\ &\leq 2\alpha_T\sqrt{2dNT\ln\Big(\frac{\sum_{t=1}^{T}\sum_{j=1}^{N}w_{u_tj}^2}{\lambda dN}+1\Big)} \end{aligned}$$

□

B Proofs of the Theorems of Section 3.1

C Proofs of the Theorems of Section 3.2

We first restate Theorem 2 in [69] regarding the confidence ellipsoid in the following Lemma.

Lemma 13 (Theorem 2 of [69])**.** *With probability at least* $1-\delta$*, the parameter* $\boldsymbol{\theta}_x^*$ *lies in the confidence ellipsoid of* $\hat{\boldsymbol{\theta}}_{x,t}$ *satisfying*

$$\|\hat{\boldsymbol{\theta}}_{x,t} - \boldsymbol{\theta}_x^*\|_{A_{x,t}} \leq \alpha_{x,t}, \forall t \geq 0$$

where $\alpha_{x,t} = R\sqrt{d_x\log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$.

Proof of Lemma 4

Proof. According to the definition of confidence interval, $CB_{v,t}(\mathbf{v}_a) = \alpha_{v,t}\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}$ and $CB_{x,t}(\mathbf{x}_a) = \alpha_{x,t}\|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$. We first prove that $\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}} \geq \|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$. By Eq (3.8), we have $\mathbf{A}_{x,t} - \lambda\mathbf{I} = \sum_{i=1}^{t} \mathbf{x}_{a_i}\mathbf{x}_{a_i}^{\mathsf{T}} = \sum_{i=1}^{t} P\mathbf{v}_{a_i}\mathbf{v}_{a_i}^{\mathsf{T}}P^{\mathsf{T}} = P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}$ and

$$\begin{aligned}\|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}} &= \sqrt{\mathbf{x}_a^{\mathsf{T}}\mathbf{A}_{x,t}^{-1}\mathbf{x}_a} \\ &= \sqrt{\mathbf{v}_a^{\mathsf{T}}P^{\mathsf{T}}\left((P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}) + \lambda\mathbf{I}\right)^{-1}P\mathbf{v}_a}.\end{aligned}$$

We can prove

$$\mathbf{v}_a^{\mathsf{T}}\mathbf{A}_{v,t}^{-1}\mathbf{v}_a \geq \mathbf{x}_a^{\mathsf{T}}\mathbf{A}_{x,t}^{-1}\mathbf{x}_a = \mathbf{v}_a^{\mathsf{T}}P^{\mathsf{T}}\left(\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\right)^{-1}P\mathbf{v}_a$$

by showing $\mathbf{A}_{v,t}^{-1} - P^{\mathsf{T}}\left(\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\right)^{-1}P$ is a positive semi-definite matrix based on the property of Schur complement.

Denote

$$M = \begin{bmatrix}\mathbf{A}_{v,t}^{-1} & P^{\mathsf{T}} \\ P & \left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\end{bmatrix}.$$

We have

$$\begin{aligned}M/\mathbf{A}_{v,t}^{-1} &= \left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I} - \left(P^{\mathsf{T}}\right)^{\mathsf{T}}\mathbf{A}_{v,t}P^{\mathsf{T}} \\ &= P\mathbf{A}_{v,t}P^{\mathsf{T}} - \lambda PP^{\mathsf{T}} + \lambda\mathbf{I} - P\mathbf{A}_{v,t}P^{\mathsf{T}} \\ &= \lambda\left(\mathbf{I} - PP^{\mathsf{T}}\right) \\ &\succeq 0\end{aligned}$$

The last step holds because P's largest singular value is smaller than 1, the eigenvalues of PP^T are smaller than 1 and $\mathbf{I} - PP^{\mathsf{T}} \succeq 0$. Because $\mathbf{A}_{v,t}^{-1} \succ 0$ and $M/\mathbf{A}_{v,t}^{-1} \succeq 0$, according to the property of Schur complement we have $M \succeq 0$. Because $\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I} = \mathbf{A}_{x,t} \succ 0$ and $M \succeq 0$, applying the property again we have $M/\left(\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\right) \succeq 0$, which gives us $\mathbf{A}_{v,t}^{-1} - P^{\mathsf{T}}\left(\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\right)^{-1}P \succeq 0$. By the definition of positive semi-definite matrix, we have $\mathbf{v}_a^{\mathsf{T}}\mathbf{A}_{v,t}^{-1}\mathbf{v}_a - \mathbf{v}_a^{\mathsf{T}}P^{\mathsf{T}}\left(\left(P(\mathbf{A}_{v,t} - \lambda\mathbf{I})P^{\mathsf{T}}\right) + \lambda\mathbf{I}\right)^{-1}P\mathbf{v}_a \geq 0$, which means $\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}} \geq \|\mathbf{x}_a\|_{\mathbf{A}_{x,t}^{-1}}$.

According to Lemma 13, $\alpha_t^v = R\sqrt{d_v \log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$ and $\alpha_t^x = R\sqrt{d_x \log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$. Since $d_v \geq d_x$, we have $\alpha_t^v \geq \alpha_t^x$. Combining the two results finishes the proof of $CB_{v,t}(\mathbf{v}_a) \geq CB_{x,t}(\mathbf{x}_a)$, which holds for any arm a at any time t. □

Proof of Lemma 5

Proof. In order to incentivize the user to pull arm a_t, the *minimum required compensation* is $\max_i \hat{r}_{x,i,t} - \hat{r}_{x,a_t,t}$. However, since the system cannot access the context features the user uses and thus maintains different reward estimates, it has to provide compensation larger than the minimum required amount.

Denote the user's greedy choice as $g = \arg\max_i \hat{r}_{x,i,t}$. To show that $c_{a_t,t}$ is sufficient, we need to prove that the user prefers the exploratory arm a_t with compensation over his/her greedy choice, i.e., $\hat{r}_{x,g,t} \leq \hat{r}_{x,a_t,t} + c_{a_t,t}$.

Based on Lemma 13, for all $t \geq 0$ with probability at least $1-\delta$, we have $|\hat{r}_{x,a,t} - \mathbb{E}[r_a]| \leq CB_{x,t}(\mathbf{x}_a)$ and $|\hat{r}_{v,a,t} - \mathbb{E}[r_a]| \leq CB_{v,t}(\mathbf{v}_a)$ hold for any arm a. Using the union bound, with probability at least $1-2\delta$ we

have

$$\begin{aligned} |\hat{r}_{x,a,t} - \hat{r}_{v,a,t}| &\leq |\hat{r}_{x,a,t} - \mathbb{E}[r_a]| + |\mathbb{E}[r_a] - \hat{r}_{v,a,t}| \\ &\leq \mathit{CB}_{x,t}(\mathbf{x}_a) + \mathit{CB}_{v,t}(\mathbf{v}_a) \end{aligned} \tag{2}$$

Then we can bound the user's reward estimate from the system side as follows,

$$\begin{aligned} \hat{r}_{x,g,t} &\leq \hat{r}_{v,g,t} + \mathit{CB}_{x,t}(\mathbf{x}_g) + \mathit{CB}_{v,t}(\mathbf{v}_g) \\ &\leq \hat{r}_{v,g,t} + 2\mathit{CB}_{v,t}(\mathbf{v}_g) \\ &\leq \hat{r}_{v,a_t,t} + 2\mathit{CB}_{v,t}(\mathbf{v}_{a_t}) \\ &\leq \hat{r}_{x,a_t,t} + \mathit{CB}_{x,t}(\mathbf{v}_{a_t}) + \mathit{CB}_{v,t}(\mathbf{v}_{a_t}) + 2\mathit{CB}_{v,t}(\mathbf{v}_{a_t}) \\ &\leq \hat{r}_{x,a_t,t} + 4\mathit{CB}_{v,t}(\mathbf{v}_{a_t}) \end{aligned} \tag{3}$$

The first and fourth steps are based on Eq (2). The second and last steps are based on Lemma 4. The third inequality is based on the UCB strategy in Eq (3.14). □

Proof of Theorem 5

Proof. Following the definition of total compensation, we have

$$\begin{aligned} \mathrm{C}(T) &= \sum_{t=1}^{T} \mathbb{E}[c_{a_t,t}] \\ &= \sum_{t=1}^{T} \left(\max_i \hat{r}_{x,i,t} - \hat{r}_{x,a_t,t} \right) \\ &\leq \sum_{t=1}^{T} \left(\max_i \left(\hat{r}_{x,i,t} + \mathit{CB}_{x,t}(\mathbf{x}_i) \right) - \hat{r}_{x,a_t,t} \right) \\ &= \sum_{t=1}^{T} \left(\hat{r}_{x,a_t,t} + \mathit{CB}_{x,t}(\mathbf{x}_{a_t}) - \hat{r}_{x,a_t,t} \right) \\ &= \sum_{t=1}^{T} \mathit{CB}_{x,t}(\mathbf{x}_{a_t}) \end{aligned}$$

where the third step holds with probability at least $1 - \delta$ and the fourth step is based on the UCB arm selection strategy.

So with probability at least $1-\delta$, we bound the total compensation as follows,

$$\begin{aligned}
\mathrm{C}(T) &\leq \sum_{t=1}^{T} CB_{x,t}(\mathbf{x}_{a_t}) \\
&\leq \sqrt{T \sum_{t=1}^{T} CB_{x,t}^2(\mathbf{x}_{a_t})} \\
&= \sqrt{T \sum_{t=1}^{T} \alpha_{x,t}^2 \|\mathbf{x}_a\|^2_{\mathbf{A}_{x,t}^{-1}}} \\
&\leq \sqrt{T \alpha_{x,T}^2 \sum_{t=1}^{T} \|\mathbf{x}_a\|^2_{\mathbf{A}_{x,t}^{-1}}} \\
&\leq \alpha_{x,T} \sqrt{T \sum_{t=1}^{T} \|\mathbf{x}_a\|^2_{\mathbf{A}_{x,t}^{-1}}}
\end{aligned}$$

According to Lemma 11 of [69], $\sum_{t=1}^{T} \|\mathbf{x}_a\|^2_{\mathbf{A}_{x,t}{}^{-1}} \leq d_x \log(\lambda + T/d_v)$. Combining with $\alpha_{x,t} = R\sqrt{d_x \log \frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$, we can complete the proof. □

Proof of Theorem 10

Proof. We bound cumulative regret by

$$\begin{aligned}
\mathrm{R}(T) &= \sum_{t=1}^{T} \left(\mathbb{E}[r_{a_t^*}] - \mathbb{E}[r_{a_t}] \right) \\
&= \sum_{t=1}^{T} \left(\mathbf{v}_{a_t^*}^{\mathsf{T}} \boldsymbol{\theta}_v^* - \mathbf{v}_{a_t}^{\mathsf{T}} \boldsymbol{\theta}_v^* \right) \\
&\leq \sum_{t=1}^{T} \left(\mathbf{v}_{a_t^*}^{\mathsf{T}} \hat{\boldsymbol{\theta}}_{v,t} + 2CB_{v,t}(\mathbf{v}_{a_t^*}) - \mathbf{v}_{a_t}^{\mathsf{T}} \boldsymbol{\theta}_v^* \right) \\
&\leq \sum_{t=1}^{T} \left(\mathbf{v}_{a_t}^{\mathsf{T}} \hat{\boldsymbol{\theta}}_{v,t} + 2CB_{v,t}(\mathbf{v}_{a_t}) - \mathbf{v}_{a_t}^{\mathsf{T}} \boldsymbol{\theta}_v^* \right) \\
&\leq \sum_{t=1}^{T} 2CB_{v,t}(\mathbf{v}_{a_t})
\end{aligned}$$

The third step holds with probability at least $1-\delta$ according to the definition of confidence interval. The fourth step holds with probability at least $1-2\delta$ according to Lemma 5, where the users are incentivized to pull arms according to UCB exploration strategy as shown in Eq (3.14). Taking a union bound, the above inequality holds with probability at least $1-3\delta$.

We continue bounding the cumulative regret with probability at least $1-3\delta$ as follows,

$$\begin{aligned}
\mathrm{R}(T) &\leq 2\sqrt{T\sum_{t=1}^{T} CB_{v,t}^2(\mathbf{v}_{a_t})} \\
&= 2\sqrt{T\sum_{t=1}^{T} \alpha_{v,t}^2 \|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}^2} \\
&\leq 2\alpha_{v,T}\sqrt{T\sum_{t=1}^{T} \|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}^2} \\
&\leq \left(2R\sqrt{d_v \log\frac{1+T/\lambda}{\delta}} + \sqrt{\lambda}\right)\sqrt{Td_v\log(\lambda+\frac{T}{d_v})}
\end{aligned}$$

where we finish the proof by combining $\sum_{t=1}^{T}\|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}^2 \leq d_v\log(\lambda+T/d_v)$ and $\alpha_{v,t} = R\sqrt{d_v\log\frac{1+t/\lambda}{\delta}} + \sqrt{\lambda}$. □

Proof of Theorem 7

Proof. With probability at least $1-2\delta$, we have

$$\begin{aligned}
\mathrm{C}(T) &\leq \sum_{t=1}^{T} 4CB_{v,t}(\mathbf{v}_{a_t}) \\
&\leq 4\sqrt{T\sum_{t=1}^{T} CB_{v,t}^2(\mathbf{v}_{a_t})} \\
&= 4\sqrt{T\sum_{t=1}^{T} \alpha_{v,t}^2 \|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}^2} \\
&\leq 4\alpha_{v,T}\sqrt{T\sum_{t=1}^{T} \|\mathbf{v}_a\|_{\mathbf{A}_{v,t}^{-1}}^2} \\
&\leq \left(4R\sqrt{d_v \log\frac{1+T/\lambda}{\delta}} + \sqrt{\lambda}\right)\sqrt{Td_v\log(\lambda+\frac{T}{d_v})}
\end{aligned}$$

□

Proof of Theorem 8

Proof. We first prove that after a fixed time point, with high probability pulling arm a once requires compensation at least $\Delta_a/3$. The proof idea is similar to the proof of Theorem 1 in [76]. We then derive the asymptotic compensation lower bound.

Based on Lemma 6, we can obtain the following inequality for all sub-optimal arms:

$$\limsup_{T\to\infty} \log(T)\|\mathbf{x}_a\|_{G_{x,T}^{-1}}^2 \leq \frac{\Delta_a^2}{2} \tag{4}$$

which is also stated in the Corollary 2 in [89].

Let $N_a(T)$ be the number of times arm a is pulled in T rounds. Since the algorithm has $o(T)$ regret, we can find $T_1'(\delta)$ such that the best arm is pulled at least $T/2$ times with probability $1-\delta/2$. Using the concentration bound we know there exists $T_1''(\delta)$ such that for $t > T_1''(\delta)$ with probability $1-\delta/2$ the confidence interval of the best arm's reward estimation is smaller than $\Delta_2/3$ where Δ_2 is the reward gap between the best arm and second best arm. Let $T_1(\delta) = \max(T_1'(\delta), T_1''(\delta))$ and for all $t > T_1(\delta)$, with probability $1-\delta$ we have $\hat{r}_{x,1,t} \geq \mathbf{E}[r_1] - \Delta_2/3$.

We argue a similar result for any suboptimal arm a. Based on Eq (4), there exists a $T_a(\delta)$ such that for any $t > T_a(\delta)$, with probability $1-\delta$

$$\|\mathbf{x}_a\|^2_{G^{-1}_{x,t}} \leq \frac{\Delta_a^2}{2\log(T)} \leq \frac{\Delta_a^2}{9 f_{T,\delta}}$$

Combining with the concentration bound in Lemma 7, we have for any $t > T_a(\delta)$ with probability $1-\delta$, $\hat{r}_{x,a,t} - \mathbf{E}[r_a] \leq \Delta_a/3$.

Let $T(\delta) = \max_i T_i(\delta)$ and we know that for any $t > T(\delta)$, the minimum required compensation to incentivize the user to pull arm a is

$$\max_i \hat{r}_{x,i,t} - \hat{r}_{x,a,t} \geq \hat{r}_{x,1,t} - \hat{r}_{x,a,t} \geq \mathbf{E}[r_1] - \frac{\Delta_2}{3} - \mathbf{E}[r_a] - \frac{\Delta_a}{3} \geq \frac{\Delta_a}{3} \tag{5}$$

with probability at least $1-\delta$.

We then use the optimization problem in Eq (3.16) to obtain the compensation lower bound, where the optimization minimizes the total compensation and satisfies the consistent constraints that the gaps of all suboptimal arms are identified with high confidence. With probability at least $1-\delta$, for sufficiently large T the total compensation is

$$C(T) \geq \sum_{a \in \mathcal{A}} \mathbf{E}[N_a(T)] \frac{\Delta_a}{3}$$

$\alpha_{\mathbf{x}_a} = \mathbf{E}[N_a(T)]/\log(T)$ is asymptotically feasible for large T because it satisfies

$$\limsup_{T\to\infty} \|\mathbf{x}_a\|^2_{H^{-1}_{x,T}} = \limsup_{T\to\infty} \log(T) \|\mathbf{x}_a\|^2_{G^{-1}_{x,T}} \leq \frac{\Delta_a^2}{2}$$

where $G_{x,T} = \log(T) H_{x,T}$. Thus for any $\epsilon > 0$, $\|\mathbf{x}_a\|^2_{H^{-1}_{x,T}} \leq \Delta_a^2/2 + \epsilon$ and

$$C(T) \geq \sum_{a \in \mathcal{A}} \mathbf{E}[N_a(T)] \frac{\Delta_a}{3} \geq c_{x,\epsilon}(\mathcal{A}, \boldsymbol{\theta}^*) \log(T) \tag{6}$$

where $c_{x,\epsilon}(\mathcal{A}, \boldsymbol{\theta}^*)$ is the the optimal value of the optimization problem in Eq (3.16) by replacing $\Delta_a^2/2$ with $\Delta_a^2/2 + \epsilon$. Since $\inf_{\epsilon>0} c_{x,\epsilon}(\mathcal{A}, \boldsymbol{\theta}^*) = c_x(\mathcal{A}, \boldsymbol{\theta}^*)$ and $T \to \infty$ we have the total compensation as

$$\Omega\left(c_x(\mathcal{A}, \boldsymbol{\theta}^*) \log(T)\right)$$

□

Remark. There are recent works on asymptotic regret lower bound for linear bandits with changing arm sets [279, 280]. It would be an interesting future work to build the asymptotic compensation lower bound with changing arm sets based on such new technique. The difference would be to construct a corresponding version of minimum required compensation (Eq (5)) for changing arm sets.

Algorithm 9 Oracle Null Space Attack

Inputs: $T, \boldsymbol{\theta}^*$
Initialize:
if Optimal objective ϵ^* of LP (4.5) > 0 **then** ▷ Attackability Test
    Find the optimal solution $\tilde{\boldsymbol{\theta}}_\perp$
    Set $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^*_\parallel + \tilde{\boldsymbol{\theta}}_\perp$
else
    return Not attackable
for $t = 1$ to T **do**
    Bandit algorithm pulls arm a_t
    Attacker observes the corresponding reward $r_t = x_{a_t}^\mathsf{T}\boldsymbol{\theta}^* + \eta_t$ from the environment
    if $x_{a_t} \neq \tilde{x}$ **then**
        Set $\tilde{r}_t = x_{a_t}^\mathsf{T}\tilde{\boldsymbol{\theta}} + \tilde{\eta}_t$ ▷ Attack
    else
        Set $\tilde{r}_t = r_t$
    Bandit algorithm observes modified reward $\tilde{r}_t$

D Proofs of the Theorems of Section 4.1

E Proofs of the Theorems of Section 4.2

We illustrate the details of oracle null space attack in Algorithm 9, which is constructed for the sufficiency proof of Theorem 13 in Section 4.2.2. We show the necessity proof of Theorem 13 below.

Necessity Proof of Theorem 13

. To prove its necessity, we will rely on the following results.

Claim 1. *Suppose arm x is pulled n times till round t by LinUCB. Its confidence bound $CB_t(x)$ in LinUCB satisfies*

$$CB_t(x) \leq O\left(\sqrt{\frac{\log t/\delta}{n}}\right). \tag{7}$$

with probability at least $1-\delta$.

Proof. By definition $\text{CB}_t(x) = \alpha_t \|x\|_{\mathbf{A}_t^{-1}}$. Denote $\mathbf{A}'_t = n \times xx^\mathsf{T}$. Since $\mathbf{A}_t = \sum_{i=1}^t x_{a_i} x_{a_i}^\mathsf{T} + \lambda\mathbf{I}$, we have $\mathbf{A}_t \succ \mathbf{A}'_t$. We can thus bound $\|x\|_{\mathbf{A}_t^{-1}}$ by

$$\|x\|_{\mathbf{A}_t^{-1}} \leq \|x\|_{\mathbf{A}_t'^{-1}} \leq \frac{L}{\sqrt{n}} \tag{8}$$

According to Theorem 2 in [69],

$$\alpha_t = \sqrt{d\log\left(\frac{1+t/\lambda}{\delta}\right)} + \sqrt{\lambda}S = O(\sqrt{\log t/\delta}).$$

Combining Eq (7) and (8) finishes the proof. □

Claim 2. *Suppose the non-target arms $\{x_a \neq \tilde{x}\}$ are pulled $o(T)$ times, the arm $\tilde{x}$ is pulled $T - o(T)$ times, and the total manipulation is C_T. With probability at least $1-\delta$, reward estimation error by the attacker*

satisfies

$$|x^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\|} - x^\mathsf{T}\boldsymbol{\theta}^*_\|| \leq \frac{C_T}{T - o(T)} + \frac{\alpha_t}{\sqrt{T - o(T)}}. \tag{9}$$

Proof.

$$\begin{aligned}
\|\hat{\boldsymbol{\theta}}_{T,\|} - \boldsymbol{\theta}^*_\|\|_2 &= \|\frac{\tilde{x}^\mathsf{T}(\hat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*)}{\|\tilde{x}\|_2^2}\tilde{x}\|_2 \\
&= \frac{1}{\|\tilde{x}\|_2^2}\|\tilde{x}^\mathsf{T}A_t^{-1}\left(\sum_{t=1}^{T} x_t(\tilde{r}_t(x_t) - x_t^\mathsf{T}\boldsymbol{\theta}^*) + \lambda\boldsymbol{\theta}^*\right)\tilde{x}\|_2 \\
&\leq \frac{1}{\|\tilde{x}\|_2^2}\|\tilde{x}^\mathsf{T}A_t^{-1}\left(\sum_{t=1}^{T} x_t\Delta_t + \sum_{t=1}^{T} x_t\eta_t + \lambda\boldsymbol{\theta}^*\right)\tilde{x}\|_2 \\
&\leq \frac{1}{\|\tilde{x}\|_2^2}\|\tilde{x}^\mathsf{T}A_t^{-1}\sum_{t=1}^{T} x_t\Delta_t\tilde{x}\|_2 + \frac{1}{\|\tilde{x}\|_2^2}\|\tilde{x}^\mathsf{T}A_t^{-1/2}\alpha_t\tilde{x}\|_2 \\
&\leq \frac{C_T}{T - o(T)} + \frac{\alpha_t}{\sqrt{T - o(T)}}
\end{aligned}$$

where the last step is because there are $T - o(T)$ of $\tilde{x}\tilde{x}^\mathsf{T}$ in A_t. We finish the proof with the fact that $\|\tilde{x}\|_2 \leq 1$. □

Now we are ready to prove that the index ϵ^* in LP (4.5) being positive is the necessary condition of an attackable environment.

Proof. Now we are ready to prove that if $\epsilon^* \leq 0$, the bandit environment is not attackable. To prove this, we show that there exists some no-regret bandit algorithm (LinUCB in particular) such that no attacking strategy can succeed. In particular, we will show that LinUCB is robust under any attacking strategy with $o(T)$ budget when $\epsilon^* \leq 0$. We prove it by contradiction: assume LinUCB is attackable with $o(T)$ budget when $\epsilon^* \leq 0$. According to Definition 4, the target arm $\tilde{x}$ will be pulled $T - o(T)$ times for infinitely many different time horizons T, and the following inequalities hold when arm $\tilde{x}$ is pulled by LinUCB:

$$\tilde{x}^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\|} + \text{CB}_T(\tilde{x}) > x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\|} + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\perp} + \text{CB}_T(x_a), \forall x_a \neq \tilde{x} \tag{10}$$

where $\hat{\boldsymbol{\theta}}_t$ is LinUCB's estimated parameter at round t based on the attacked rewards. We decompose $\hat{\boldsymbol{\theta}}_T = \hat{\boldsymbol{\theta}}_{T,\|} + \hat{\boldsymbol{\theta}}_{T,\perp}$, where $\tilde{x} \perp \hat{\boldsymbol{\theta}}_{t,\perp}$ and $\tilde{x} \parallel \hat{\boldsymbol{\theta}}_{T,\|}$. We will now show that the above inequalities lead to

$$\tilde{x}^\mathsf{T}\boldsymbol{\theta}^*_\| > x_a^\mathsf{T}\boldsymbol{\theta}^*_\| + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\perp}, \forall x_a \neq \tilde{x}$$

when $T \to \infty$.

By Claim 1 we have

$$\text{CB}_T(\tilde{x}) \leq O\left(\sqrt{\frac{\log T/\delta}{T - o(T)}}\right)$$

We also have

$$\text{CB}_T(x_a) = \alpha_T\|x_a\|_{\mathbf{A}_T^{-1}} > 0$$

By Claim 2 we have

$$x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\|} \geq x_a^\mathsf{T}\boldsymbol{\theta}_\|^* - \frac{C_T}{T-o(T)} - \frac{\alpha_T}{\sqrt{T-o(T)}}$$
$$\tilde{x}^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\|} \leq \tilde{x}^\mathsf{T}\boldsymbol{\theta}_\|^* + \frac{C_T}{T-o(T)} + \frac{\alpha_T}{\sqrt{T-o(T)}}$$

Substitute them back and we have that with probability at least $1-3\delta$,

$$\tilde{x}^\mathsf{T}\boldsymbol{\theta}_\|^* > x_a^\mathsf{T}\boldsymbol{\theta}_\|^* + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\perp} + \mathrm{CB}_T(x_a) - O\left(\sqrt{\frac{\log T/\delta}{T-o(T)}}\right) - \frac{2C_T}{T-o(T)} - \frac{2\alpha_T}{\sqrt{T-o(T)}}, \forall x_a \neq \tilde{x}$$

Taking $T \to \infty$ and noticing that $C_T = o(T)$, the last three terms on the right-hand side diminish to 0 and we have,

$$\tilde{x}^\mathsf{T}\boldsymbol{\theta}_\|^* > x_a^\mathsf{T}\boldsymbol{\theta}_\|^* + x_a^\mathsf{T}\hat{\boldsymbol{\theta}}_{T,\perp}, \forall x_a \neq \tilde{x}$$

This implies that there must exist a $\hat{\boldsymbol{\theta}}_{T,\perp}$ that satisfies $\tilde{x} \perp \hat{\boldsymbol{\theta}}_{T,\perp}$ and makes the objective of LP (4.5) larger than 0. Therefore, its optimal objective ϵ^* must also be positive. This however contradicts our assumption $\epsilon^* \leq 0$, implying that LinUCB is not attackable by any attack strategy. □