Structured Bandits and Applications

Exploiting Problem Structure for Better Decision-making under Uncertainty

by

Sharan Vaswani

B.E., Birla Institute of Technology and Science, Pilani, 2012
M.Sc., The University of British Columbia, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies
(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

February 2019

Abstract

We study the problem of decision-making under uncertainty in the bandit setting. This thesis goes beyond the well-studied multi-armed bandit model to consider structured bandit settings and their applications. In particular, we learn to make better decisions by leveraging the application-specific problem-structure in the form of features or graph information. We investigate the use of structured bandits in two practical applications: online recommender systems with an available network of users and viral marketing in social networks. For each of these applications, we design efficient bandit algorithms and theoretically characterize their performance. We experimentally evaluate the efficiency and effectiveness of these algorithms on real-world datasets. For applications that require modelling complex non-linear feature-reward relationships, we propose a bootstrapping approach and establish theoretical regret bounds for it. Furthermore, we consider the application of multi-class classification with bandit feedback as a test-bed for evaluating our bootstrapping approach.

Lay Summary

Making decisions under partial or incomplete information is important in applications ranging from clinical trials to computational advertising and marketing. This work maps modern applications such as recommender systems and viral marketing in social networks to the traditional framework for decision-making under uncertainty. It leverages the application-specific problem-structure in order to design scalable and theoretically sound algorithms. These algorithms provably learn to make better decisions by repeated interaction with the system at hand. We also experimentally demonstrate the effectiveness and efficiency of our approach. Beyond these specific applications, we propose and analyse general algorithms that enable us to make better decisions in complex scenarios that require the expressivity of modern machine learning models.

Preface

The main matter of this thesis is based on papers that have either been published or are under review:

- Chapter 2 is the result of two publications: (Wen et al., 2017) that appeared in Neural Information Processing Systems, 2017:
  Wen, Zheng, Branislav Kveton, Michal Valko, and Sharan Vaswani. "Online influence maximization under independent cascade model with semi-bandit feedback." In Advances in Neural Information Processing Systems, 2017.
  and (Vaswani et al., 2017a) that appeared in the International Conference on Machine Learning, 2017:
  Vaswani, Sharan, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. "Model-independent online learning for influence maximization." In International Conference on Machine Learning, 2017.

  For (Wen et al., 2017), the proofs were mainly done by Zheng Wen, Branislav Kveton and Michal Valko whereas the experiments were done by Sharan Vaswani. The majority of the paper's content was written by Zheng Wen and Sharan Vaswani. The paper (Vaswani et al., 2017a) was mainly written by Sharan Vaswani who is also the major contributor to the experimental section. The theoretical results were contributed jointly by Sharan Vaswani, Zheng Wen, Branislav Kveton and Mohamed Ghavamzadeh. Valuable feedback on the paper was provided by Laks Lakshmanan and Mark Schmidt.

  We note that this work also constitutes a US patent titled "Influence Maximization Determination in a Social Network System", co-authored by Sharan Vaswani, Zheng Wen, Branislav Kveton and Mohamed Ghavamzadeh, and filed by Adobe Research in August, 2017.

- Chapter 3 was published in the International Conference on Artificial Intelligence and Statistics, 2017 (Vaswani et al., 2017b):
  Vaswani, Sharan, Mark Schmidt, and Laks Lakshmanan. "Horde of Bandits using Gaussian Markov Random Fields." In Artificial Intelligence and Statistics,2017.

  Sharan Vaswani is the major contributor on this paper in terms of the writing, theoretical results and experimental evaluation. The work was supervised by Mark Schmidt and Laks V.S. Lakshmanan who provided useful feedback and helped in refining the paper's content.

- Chapter 4 consists of the work done in (Vaswani et al., 2018) that is under submission and is available as a preprint:
  Vaswani, Sharan, Branislav Kveton, Zheng Wen, Anup Rao, Mark Schmidt, and Yasin Abbasi-Yadkori. "New Insights into Bootstrapping for Bandits." arXiv preprint arXiv:1805.09793, 2018.

  This paper was mainly written by Sharan Vaswani who also performed the experiments. Zheng Wen, Sharan Vaswani and Branislav Kveton are equal contributors to the theoretical results. The paper was constantly discussed and refined with the help of Anup Rao, Mark Schmidt and Yasin Abbasi-Yadkori.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This work has been possible because of the help and kindness of numerous individuals throughout the last three years. I am especially grateful to my supervisors, Mark Schmidt and Laks Lakshmanan for their constant help and support. They encouraged me to think independently, gave me the freedom to pursue what I became interested in and ensured that I do not stray wildly off track in my research. From them, I have not only learnt important technical material in machine learning, data mining and optimization, but also the manner in which research should be conducted. I have been fortunate enough to have a set of “unofficial” advisors - Branislav Kveton and Zheng Wen at Adobe Research. A large part of this thesis has been done in collaboration with them. I thank them for teaching me the basics of bandits, being patient with my questions and for showing me how to truly enjoy research.

I would like to thank other co-authors who contributed to this thesis: Mohammad Ghavamzadeh, Michal Valko, Yasin Abassi-Yadkori and Anup Rao. I really hope that we continue to work together in the future. I would like to thank all the professors who taught me at UBC and were more than willing to answer my questions, in particular, Nick Harvey, Hu Fu and Siamak Ravanbaksh. I am grateful to Alex Bouchard for agreeing to serve on the PhD committee and for timely and valuable feedback. Thank you to Rachel Pottinger and Will Welch for serving as the university examiners and to Csaba Szepesvári for agreeing to be the external examiner of this thesis.

A grateful thank you to all the staff: Joyce Poon, Kath Imhiran and Lara Hall for all their help in navigating the administrative jungle. I would also like to thank UBC for the Four Year Fellowship that mainly funded my PhD. I am grateful to all the people who let me intern and work with them: Limespot for giving me the valuable experience of consulting for them, Branislav Kveton, Zheng Wen and Mohammad Ghavamzadeh at

Adobe Research, Hoyt Koepke and Krishna Sridhar at Apple and Francis Bach at Inria.

The PhD journey is a stressful one, one that can not be undertaken without the strong support of friends and family. I would like to thank all my lab-mates for our technical discussions and not-so-technical rants. A special thanks to Prithu, Glenn, Zainab, Nasim, Reza, Alireza, Jason, Julie, Issam and Mohamed. I have been lucky to have found a great group of friends throughout my stay at UBC. A big thanks to Aarya, Abhinav, Farooq, Chandan, Yasasvy and Manav for keeping me sane through these years, for all the hikes and road-trips and Tim-Hortons catch-ups, for patiently listening to me complain and helping me move houses every year! My internships and work experiences have been made enjoyable by the great company I found: thank you Sampoorna, Stefano, Mara, Lakhan, Params, Almas, Shreyas, Lars, Thomas, Leanic, Ignacio, Raghu and Polina. Thank you to the friendly staff at Bean Around the World on campus for letting me sit and work for hours together and for surprise free coffee.

Most importantly, I would like to thank my family for their constant support. A special thank you to my sister, Smriti, for valuable advice and chiding in our regular phone conversations that always manage to make me feel better. Finally, I would like to dedicate this thesis to my mother for always believing in me, listening to my ambiguous rants about research and for showing by example that “Although it is nice to be important, it is more important to be nice.” I am particularly grateful to her for patiently teaching me middle-school algebra without which this thesis would not have been possible!

To Mumma,
for her unconditional love and support...

Chapter 1

Introduction

Numerous applications require making a sequence of decisions under partial or incomplete information. For example, consider a rover exploring the surface of Mars. Such a rover does not have complete information about the Mars terrain and needs to make decisions about its navigation on the planet. In the classic decision-making under uncertainty framework, the rover is referred to as the *agent* making decisions, the Mars surface corresponds to the *environment* in which decisions are made and the actuators or controls used in the navigation are termed as *actions*. Let us assume that the aim of the rover is to find water on Mars and it needs to make a sequence of decisions about its navigation (for example: turn left, walk ahead 10 steps) to achieve this. After each decision, the rover receives *feedback* from the environment using its sensors; for instance, it can detect if it is in the vicinity of a potential water source or if it has moved too close to a crater. This interaction consisting of a decision and its corresponding feedback is referred to as a *round*. In this framework, the agent's decision is associated with a *reward* model. In our example, the agent receives a high reward if it finds a potential source of water, detects methane in the atmosphere or even explores the Mars surface safely. The aim of the agent is to maximize its cumulative reward across rounds by making decisions based on the *history* of actions, feedback and rewards.

An important special case of the above framework is known as the *bandit* setting and encompasses applications such as clinical trials (Thompson, 1933), A/B testing (Agarwal et al., 2016), advertisement placement (Tang et al., 2013), recommender systems (Li et al., 2010) and network routing (Gai et al., 2012). For instance, in a clinical trial, the aim is to infer the “best” drug (for example, one that has the least side-effects) amongst a set of available drugs. Let us map a clinical trial to the above framework; the agent corresponds to the clinician running the trial whereas the environment consists of the set of patients to which the drugs will be administered. In this case, a patient arrives in each round and the action consists of administering a particular drug to them. The feedback is the effectiveness of the drug in curing a patient and what side-effects it leads to. The cumulative reward across rounds corresponds to the number of patients that were cured without any major side-effects. The bandit setting and its applications will be the main focus of this thesis and we describe it in detail in the next section.

1.1 Multi-armed Bandits

The multi-armed bandit (MAB) framework (Lai and Robbins, 1985; Bubeck and Cesa-Bianchi, 2012; Auer, 2002; Auer et al., 2002) consists of *arms* that correspond to different decisions or actions. These may be different treatments in a clinical trial or different products that can be recommended to a user of an online service. The generic protocol followed in a MAB framework can be summarized in Algorithm 1. The protocol consists of T rounds. In each round, the agent uses a bandit algorithm in order to *select* an arm to “pull”. Pulling an arm is equivalent to taking the action corresponding to that arm.

Once an arm is pulled, the agent *observes* a reward and corresponding feedback from the environment. A key feature of the bandit framework is that we observe the feedback *only* for the arm(s) that have been pulled in a given round. Finally, the agent *updates* its estimate of the arm's reward or the action's utility. An example of a simple bandit algorithm would be the “greedy” strategy where the agent selects the arm with the highest reward obtained thus far. In the clinical trial example, the estimated mean reward for a drug can be the proportion of people that were cured by using that particular drug.

Algorithm 1 Generic Bandit Framework

```
for $t = 1$ to $T$ do
    SELECT: Use the bandit algorithm to decide which arm(s) to pull.
    OBSERVE: Pull the selected arm(s) and observe the reward and associated feedback.
    UPDATE: Update the estimated reward for the arms(s).
```

As explained before, the aim of the agent is to select and pull the arms that maximize the cumulative reward across the T rounds. We now describe the above protocol in detail.

Depending on the assumptions about the reward, the MAB framework can be classified into the *stochastic* (Auer et al., 2002) or *adversarial* setting (Auer et al., 1995). In this thesis, we exclusively focus on stochastic multi-armed bandits. In the stochastic setting, each arm has an associated reward distribution and every pull of an arm corresponds to sampling its corresponding distribution. The mean of this distribution is equal to the *expected reward* or utility of pulling that arm. The stochastic MAB setting models random independent but identically distributed fluctuations in an arm's mean reward. For example, in a clinical trial, the patients are independent of each other and assumed to be homogeneous. The reward from each pull of an arm (administration of a drug) can thus be modelled as an independent random variable from an underlying distribution. Similarly, in a recommender system, the reward for pulling an arm (equivalent to recommending the corresponding product to a user) is equal to the rating it receives from the user.

Notice that if we had complete information about each arm's expected utility, the optimal decision is to always pull the arm with highest expected utility, thus maximizing the cumulative reward in expectation. In the absence of this information, the agent learns to infer the utility of the arms by repeatedly interacting with the system in the trial-and-error fashion described in Algorithm 1. Note that the MAB framework can be generalized to account for some auxiliary feedback from the pulled arm. For example, in the recommender system scenario, a user review can be viewed as such auxiliary feedback. This additional feedback can be used to better discriminate the "good" (with higher rewards) arms from the "bad" sub-optimal ones.

The agent's aim of maximizing the cumulative reward across rounds results in an *exploration-exploitation trade-off*. Here, *exploration* means choosing an arm to gain more information about it, while *exploitation* corresponds to choosing the arm with the highest reward given the agent's past observations. For example, in the context of news-recommendation, exploration seeks to learn more about a user's preferences on a topic of news they haven't encountered before, while exploitation means recommending the news topic that the system believes (given the user's past history) that they will like the most.

We now give a formal problem definition of the stochastic MAB problem. We denote the number of arms by K and index them with j. The expected reward on pulling the arm j is given by μ_j. We denote the index of the arm pulled in round t as j_t. After pulling this arm, the agent receives a reward r_t sampled from the arm's underlying reward distribution, in particular, $\mathbb{E}[r_t | j_t = j] = \mu_j$. We define the *best* or optimal arm as the one with the highest expected reward. The objective is to maximize the expected cumulative reward - the reward accumulated across rounds in expectation. An equivalent objective is to minimize the *expected cumulative regret.* The cumulative regret $R(T)$ is the cumulative loss in the reward across T rounds because of the lack of knowledge of the optimal arm. In the MAB setting, the expected cumulative regret is defined as follows:

$$\mathbb{E}[R(T)] = T \max_j \mu_j - \sum_{t=1}^{T} \mu_{j_t} \tag{1.1}$$

1.2 Structured Bandits

The MAB framework assumes the arms to be independent of each other and can not share information between them. This assumption is often too restrictive in practical applications where the number of arms can be large. For example, in the context of a recommender system where each arm corresponds to a product that can be recommended, it is useful to use the feedback for a recommended item to infer the user's preference on a similar item (arm) that was not recommended. Consequently, substantial recent research (Dani et al., 2008; Li et al., 2010; Filippi et al., 2010b; Riquelme et al., 2018) considers additional information in the form of features for the arms. These features might describe additional information about a drug's constituents or can correspond to the description of a product to be recommended. The similarity between arms can then be captured by their proximity in the feature space.

Previous work uses a parametric model to map the arms' features to their expected rewards. Most of this work (Dani et al., 2008; Rusmevichientong and Tsitsiklis, 2010; Abbasi-Yadkori et al., 2011; Li et al., 2010; Agrawal and Goyal, 2013b) considers the *linear bandit* setting. This setting assumes that the expected reward for an arm is given by the inner product of the arm's features and an underlying unknown parameter vector. Formally, if d is the dimensionality of the feature space and $\mathbf{x}_j \in \mathbb{R}^d$ are the features corresponding to arm j, then its expected reward is given as: $\mu_j = \langle \mathbf{x}_j, \theta^* \rangle$. Here, θ^* is the unknown d-dimensional vector mapping the features to the reward and needs to be inferred from the past observations. Alternatively, the linear bandit setting corresponds to a Gaussian reward distribution with mean μ_j and a known variance σ^2. If $d = K$ and the features are standard basis vectors, then this framework becomes equivalent to the traditional MAB setting considered in the previous section.

For linear bandits, the definition of the best-arm and the cumulative regret is the same as in the MAB case. In particular, the expected cumulative regret $\mathbb{E}[R(T)]$ is given as:

$$\mathbb{E}[R(T)] = T \max_{j} \left[\langle \mathbf{x}_j, \theta^* \rangle\right] - \sum_{t=1}^{T} \langle \mathbf{x}_{j_t}, \theta^* \rangle \tag{1.2}$$

The above setting can be generalized to the *contextual bandit* setting (Langford and Zhang, 2008; Li et al., 2017; Chu et al., 2011) in which the arms' feature vectors can vary across the rounds. The contextual bandit setting can be used to model the news-recommendation scenario. The varying features correspond to the content of constantly changing news articles and the underlying vector θ^* can be used to model a particular user's preferences for different news categories. In this setting, the best-arm also varies with the round and depends on the set of arm features in that particular round. We refer to this set of features as *context vectors* and denote it by $\mathcal{C}_t = [\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \ldots \mathbf{x}_{K,t}]$. Given this definition, the expected cumulative regret can be given as:

$$\mathbb{E}[R(T)] = \sum_{t=1}^{T} \left[\max_{\mathbf{x} \in \mathcal{C}_t} \left(\langle \theta^*, \mathbf{x} \rangle \right) - \langle \theta^*, \mathbf{x}_{j_t,t} \rangle \right] \tag{1.3}$$

Another generalization of the linear bandit setting involves using complex non-linear mappings in order to model the feature-reward relationship. For example, expressive non-linear mappings such as neural networks can better capture the relationship between the expected reward and the content in product descriptions or news articles. Filippi et al. (2010a) use generalized linear models, McNellis et al. (2017) consider decision trees and recently, Riquelme et al. (2018) used deep neural networks in order to model the feature-reward function. In this setting, the expected reward of pulling arm j is given as: $\mu_j = m(\mathbf{x}_j, \theta^*)$. Here the function $m(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is referred to as the *model class* and may refer to a neural network or a generalized linear model.

Note that the lower bound on the achievable cumulative regret for the MAB setting is $\Omega(\sqrt{KT})$ (Auer et al., 2002) whereas it is of the order of $\Omega(\sqrt{dT})$ in the linear bandit setting (Dani et al., 2008) where d is the feature dimension. In Section 1.3, we survey bandit algorithms that achieve this lower bound on the regret.

We now consider the UPDATE step in the generic Algorithm 1. Given that we have had t rounds of interaction, we need to maintain the estimated mean reward for each of the arms. In the MAB case, the estimated mean reward for an arm is simply the average of the rewards observed when that particular arm has been pulled thus far. For structured bandits, let us first consider the update in the general non-linear case. At round t, the *history* of interactions can be described by the set of features (corresponding to the pulled arms) and the feedback obtained thus far. In particular, let $\mathbf{x}_i$ be the features corresponding to the arm pulled at round i and let y_i be its corresponding scalar reward[1]. Let $\mathcal{D}_t = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_t, y_t)\}$ be the set of features and rewards obtained until round t.

Assuming the generative model from features to rewards is parametrized by the vector θ, the log-likelihood of observing the data $\mathcal{D}_t$ is given by $\mathcal{L}_t(\theta) = \sum_{i \in \mathcal{D}_t} \log\left[\mathcal{P}(y_i | x_i, \theta)\right]$ where $\mathcal{P}(y_i | x_i, \theta)$ is the probability of observing label y_i given the feature vector $\mathbf{x}_i$, under the model parameters θ. In the MAB case without features, the probability of observing y_i (for all $i \in [t]$) is simply given by $\mathcal{P}(y_i | \theta)$. The maximum likelihood estimator (MLE) for the observed data is defined as $\widehat{\theta}_t \in \arg\max_\theta \mathcal{L}_t(\theta)$. If $m(\cdot, \cdot)$ is the model class, the estimated mean reward at round t for an arm with feature vector $\mathbf{x}$ is given as $m(\mathbf{x}, \widehat{\theta}_t)$.

In the linear bandit case, the MLE $\widehat{\theta}_t$ can be computed in a closed form. If X_t is the $t \times d$ matrix of features and $\mathbf{y}_t$ is the t-dimensional vector of observations obtained until round t, then $\widehat{\theta}_t$ is given by:

$$\widehat{\theta}_t = (X_t^T X_t)^{-1} X_t^T y_t \tag{1.4}$$

In this case, the estimated mean reward for an arm with features $\mathbf{x}$ is equal to the inner product $\langle \mathbf{x}, \widehat{\theta}_t \rangle$.

In the next section, we consider the SELECT phase of Algorithm 1 and briefly survey the bandit algorithms that trade off exploration and exploitation.

1.3 Algorithms

For both the MAB and contextual bandit settings, there are three main strategies for addressing the exploration-exploitation tradeoff: (i) ε-greedy (Langford and Zhang, 2008)

[1] We use the notation $\mathbf{x}_i$ to refer to the features for point i in the history and $\mathbf{x}_j$ to denote the features for arm j. These are used mutually exclusively and should be clear from the context.

(ii) Optimism in the Face of-Uncertainty (Auer, 2002; Abbasi-Yadkori et al., 2011) and (iii) Thompson sampling (Agrawal and Goyal, 2013b). We briefly describe these algorithms in the linear bandit setting since MAB is a special case of this setting.

- **ε-Greedy** explicitly differentiates between the rounds in which to either explore or exploit. In each round, the algorithm chooses to explore with probability equal to ε. In an exploration round, the agent chooses an action uniformly at random. While exploiting, it chooses the action with the maximum estimated mean reward at that round. This can be expressed formally in the linear bandit setting as follows:

$$j_t \sim \text{Uniform}\{1, 2, \dots K\} \qquad \text{(With probability } \varepsilon\text{)}$$
$$j_t = \arg\max_j \langle \mathbf{x}_j, \widehat{\theta}_t \rangle \qquad \text{(With probability } 1 - \varepsilon\text{)}$$

  If the parameter ε is chosen correctly, ε-Greedy results in a sub-linear but sub-optimal $O(T^{2/3})$ regret bound (Langford, 2007). In practice, it is difficult to set the ε parameter and the algorithm's performance is sensitive to this choice.

- **Optimism in the Face of Uncertainty** (OFU) based algorithms (Auer et al., 2002) (Abbasi-Yadkori et al., 2011) address the exploration-exploitation trade-off in an optimistic fashion by choosing the arm that maximizes the *upper confidence bound*. As the name suggests, the upper confidence bound (UCB) is an upper bound on the expected reward for an arm in a particular round. Mathematically, it can be written as a non-negative linear combination of the estimated mean reward and its standard deviation. Formally, in the linear bandit case, the algorithm chooses the arm j_t according to the following rule:

$$j_t = \arg\max_j \left[\langle \mathbf{x}_j, \widehat{\theta}_t \rangle + c \cdot \sqrt{\mathbf{x}_j^\intercal M_t^{-1} \mathbf{x}_j} \right] \tag{1.5}$$

  Here M_t is the d-dimensional covariance matrix equal to $X_t^T X_t$. The first term corresponds to the mean reward in round t, the second term is the standard deviation and c (≥ 0) is the trade-off parameter. Maximizing the first term corresponds to exploitation whereas the second term is large for arms that haven't been explored enough. The trade-off parameter c is determined theoretically and decreases with t, thus favouring exploitation after all the arms have been explored sufficiently. If c is chosen appropriately, UCB has been proved to attain the near-optimal $\widetilde{O}(\sqrt{T})$ regret in the MAB (Auer et al., 2002), linear bandit (Dani et al., 2008; Abbasi-Yadkori et al., 2011) and generalized linear bandit (Filippi et al., 2010b) settings. Here, the $\widetilde{O}(\cdot)$ notation suppresses additional log factors.

- **Thompson sampling** (Thompson, 1933) is a bandit algorithm popularly used in practice. The algorithm assumes a prior distribution on the parameters θ and forms the posterior distribution $\mathcal{P}(\theta|\mathcal{D}_t)$ given the observations until round t. It obtains a sample $\widetilde{\theta}$ from the posterior and then chooses the arm maximizing the reward conditioned on this sample. Formally, Thompson sampling (TS) can be described as follows:

$$\begin{aligned}&\widetilde{\theta} \sim \mathcal{P}(\theta|\mathcal{D}_t)\\ &j_t = \arg\max_j \langle \mathbf{x}_j, \widetilde{\theta}\rangle\end{aligned} \tag{1.6}$$

  In the linear bandit case, both the prior and the posterior distributions are Gaussian and obtaining the sample $\widetilde{\theta}$ is computationally efficient. TS uses the variance in the sampling procedure to induce exploration and has been shown to attain an $\widetilde{O}(d\sqrt{T})$ regret in the MAB (Agrawal and Goyal, 2012a), linear (Agrawal and Goyal, 2012b) and generalized linear (Abeille and Lazaric, 2016) bandit settings.

1.4 Summary of contributions

In this work, we focus on the applications of structured bandits to problems in viral marketing in social networks and recommender systems. Our list of contributions is as follows:

- **Chapter 2**: We show how the linear bandit framework can be used for viral marketing in social networks. We focus on the problem of *influence maximization* (IM) in which an agent aims to learn the set of "best influencers" in a social network online while repeatedly interacting with it.

  In the first part of this chapter, we study this problem under the popular *independent cascade* model of information diffusion in a network. Under a specific feedback model, we propose and analyse a computationally efficient *upper confidence bound*-based algorithm. Our theoretical bounds on the cumulative regret achieve near-optimal dependence on the number of interactions and reflect the *topology* of the network and the *activation probabilities* of its edges, thereby giving insights on the problem complexity. To the best of our knowledge, these are the first such results.

In the second part of this chapter, we propose a novel reparametrization for the above problem that enables our framework to be agnostic to the underlying model of diffusion. It also allows us to use a weaker model of feedback from the network, while retaining the ability to learn in a statistically efficient manner. We design an upper confidence bound algorithm and theoretically analyse it. Experimentally, we show that our framework is robust to the underlying diffusion model and can efficiently learn a near-optimal solution.

- **Chapter 3**: We show how to leverage the contextual bandit framework and additional side-information in the form of a social network in an online content-based recommender system. We exploit a connection to Gaussian Markov Random Fields in order to make our approach scalable and practically viable for large real-world networks. We prove regret bounds for variants of the ε-greedy and Thompson sampling algorithms. We show the effectiveness of our approach by systematic experiments on real-world datasets.

- **Chapter 4**: We propose to use bootstrapping for addressing the exploration-exploitation trade-off for complex non-linear feature-reward mappings. We first show that the commonly used non-parametric bootstrapping procedure can be provably inefficient and establish a near-linear lower bound on the regret incurred by it under the bandit model with Bernoulli rewards. As an alternative, we propose a *weighted bootstrapping* (WB) procedure. We show that for both Bernoulli and Gaussian rewards, WB is mathematically equivalent to Thompson sampling and results in near-optimal regret bounds. These are the first theoretical results for bootstrapping in the context of bandits. Beyond these special cases, we show that WB leads to better empirical performance than TS for several reward distributions bounded on $[0, 1]$. For the contextual bandit setting, we give practical guidelines that make bootstrapping simple and efficient to implement. We show that it results in good empirical performance on a multi-class classification task with bandit feedback.

Chapters 2- 4 have corresponding appendices that contain full proofs of the theoretical results and additional experimental results. In Chapter 5, we discuss some future directions and extensions of the work in this thesis.

Chapter 2

Influence Maximization

In this chapter, we consider the application of linear bandits to the problem of influence maximization in social networks.

2.1 Introduction

Social networks have become increasingly important as media for spreading information, ideas and influence. For instance, social media campaigns play a significant role in promoting and publicizing movies, concerts or recently released products. These campaigns rely on *viral marketing* to spread awareness about a specific product via word-of-mouth propagation over a social network. There have been numerous studies (Kempe et al., 2003; Easley and Kleinberg, 2010; Myers and Leskovec, 2012; Gomez-Rodriguez and Schölkopf, 2012; Gomez Rodriguez et al., 2013) characterizing the factors influencing information diffusion in such networks.

A particular setting in viral marketing is the *influence maximization* (IM) (Kempe et al., 2003; Chen et al., 2013a) problem. In this setting, marketers aim to select a fixed number of influential users (called *seeds*) and provide them with free products or discounts. They assume that these users will influence their neighbours and, transitively, other users in the social network to adopt the product. This will result in information propagating across the network as more users adopt or become aware of the product. The marketer has a budget on the number of free products that can be given. They must thus choose seeds strategically in order to maximize the *influence spread*, which is the expected number of users that become aware of the product.

Existing solutions (Chen et al., 2009; Leskovec et al., 2007; Goyal et al., 2011b,a; Tang et al., 2014, 2015b) to the IM problem require as input, the social network and the underlying diffusion model that describes how information propagates through the network. The social network is modelled as a directed graph with the nodes representing users, and the edges representing relations (e.g., friendships on Facebook, followers on Twitter) between them. The IM problem has been studied under various probabilistic diffusion models such as the Independent Cascade (IC) and Linear Threshold (LT) models (Kempe et al., 2003). These common models are parametrized by *influence probabilities* that correspond to the edge weights of the corresponding graph. In other words, each directed edge (i, j) is associated with an influence probability that models the strength of influence that user i has on user j.

Knowledge of the underlying diffusion model and its parameters is essential for the existing IM algorithms to perform well. For instance, in (Goyal et al., 2011a), the authors showed that even when the diffusion model is known, correct knowledge of the model parameters is critical to choosing "good" set of seeds that result in a large spread of influence. In many practical scenarios, however, the influence probabilities are *unknown.* Some papers set these heuristically (Chen et al., 2010; Yadav et al., 2017) while others try to learn these parameters from past propagation data (Saito et al., 2008; Goyal et al., 2010; Netrapalli and Sanghavi, 2012). However in practice, such data is difficult to obtain and the large number of parameters (of the order of network size) makes this learning challenging. Another challenge is the choice of the model that best captures the characteristics of the underlying diffusion. In practice, it is not clear how to choose from amongst the increasing number of plausible diffusion models (Kempe et al., 2003; Gomez Rodriguez et al., 2012; Li et al., 2013). Furthermore, in (Du et al., 2014), the authors empirically showed that misspecification of the diffusion model can lead to choosing highly sub-optimal seeds, consequently making the IM campaign ineffective.

These concerns motivate the learning framework of IM bandits (Vaswani et al., 2015; Valko, 2016; Chen et al., 2016a). In this framework, the marketer conducts independent influence maximization campaigns across multiple rounds. Each round corresponds to a campaign for the same or similar products. The aim of the marketer is to learn the factors influencing the diffusion and use this knowledge to design more effective marketing campaigns. For example, this trial and error procedure might reveal that a particular product is not popular among certain demographics; the marketer can then correct for this in subsequent IM campaigns. This problem can be mapped to the generic bandit framework of Algorithm 1 as follows: the agent corresponds to the marketer, the environment to the network and a possible action corresponds to choosing a set of users as seeds. In each round, the marketer chooses a seed set (SELECT), receives feedback (OBSERVE) from the network and utilizes this information to better estimate (UPDATE) the diffusion process.

Depending on the feedback received about the diffusion, IM bandits (IMB) can be classified as follows: (i) Full-bandit feedback, where only the *number of influenced nodes* is observed; (ii) Node semi-bandit feedback, where the *identity of influenced nodes* is observed; or (3) Edge semi-bandit feedback, where the *identity of influenced edges* (edges along which the information diffused) is also observed. In this work, we will mainly consider the edge semi-bandit feedback model and a relaxed version of it. We argue that it is reasonable to obtain such feedback in practical scenarios; for example, it is easy for Facebook to infer which of your friends influenced you to share a particular article. Similarly, E-commerce companies can keep track of users that referred a particular product to their peers. In both these cases, it is possible to trace the precise path along which information diffused.

Similar to the general bandit setting introduced in Chapter 1, the aim of the marketer is to minimize the loss in the influence spread because of the lack of knowledge about the diffusion process. Here, exploration consists of choosing seeds that improve the marketer's knowledge of the diffusion process; whereas exploitation corresponds to choosing a seed set that is estimated to have a large expected spread.

From a bandits perspective, IMB combines two main challenges: first, the number of actions (number of possible seed sets that can be selected) grows exponentially with the cardinality of the set. Second, we only observe the influenced portion of the network. This limits the feedback received in each round, making the learning problem difficult. In this chapter, we address these challenges in two different settings. We first set up the necessary notation and give a more formal problem definition in Section 2.2.

In Section 2.3, we study the IMB problem under the independent cascade model and edge semi-bandit feedback. In Section 2.3.4, we identify *complexity metrics* that capture the information-theoretic complexity for the IMB problem. We then propose `ICLinUCB`, a `UCB`-like algorithm (Section 2.3.3) suitable for large-scale IM problems and bound its cumulative regret in Section 2.3.4. Our regret bounds are polynomial in all quantities of interest and have near-optimal dependence on the number of interactions. They reflect the structure and activation probabilities of the network and do not depend on inherently large quantities, such as the reciprocal of the minimum probability of being influenced (unlike (Chen et al., 2016a)) or the exponential size of the space of actions. Finally, we evaluate `ICLinUCB` on several problems in Section 2.3.5. We perform experiments on simple representative topologies and empirically show that the regret of `ICLinUCB` scales as suggested by our topology-dependent regret bounds. We also show that `ICLinUCB` with linear generalization can lead to low regret in real-world online influence maximization problems.

In Section 2.4, we first propose a novel parametrization for the IM problem that enables our bandit framework to be agnostic to the underlying model of diffusion. This formulation in Section 2.4.1 lets us relax the requirement for edge semi-bandit feedback to a weaker notion which we term as *pairwise reachability feedback*.[1] Under this feedback model, we formulate IMB as a linear bandit problem, propose a scalable LinUCB-based algorithm (Section 2.4.4) and bound its cumulative regret in Section 2.4.5. We show that our regret bound has the optimal dependence on the time horizon, is linear in the cardinality of the seed set, and has a better dependence on the size of the network as compared to the previous literature. Finally, we give some implementation guidelines in Section 2.4.6. In Section 2.4.7, we empirically evaluate our proposed algorithm on a real-world network and show that it is statistically efficient and robust to the underlying diffusion model.

Finally, we survey the related work in Section 2.5 and conclude by giving some directions for future work in Section 2.6.

2.2 Problem Formulation

Recall that the social network is modelled as directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set $\mathcal{V}$ of $n = |\mathcal{V}|$ nodes (users), the set $\mathcal{E}$ of $m = |\mathcal{E}|$ directed edges (user relations) and edge weights (influence probabilities) given by function $p : \mathcal{E} \to [0, 1]$. Throughout this work, we use $p(e)$ and $p_{u,v}$ interchangeably to refer to the influence probability for the edge e between nodes u and v. The IM problem is characterized by $(\mathcal{G}, p, \mathcal{C}, \mathcal{D})$, where $\mathcal{C}$ is the collection of feasible seed sets, and $\mathcal{D}$ is the underlying diffusion model. The collection of feasible seed sets $\mathcal{C}$ is determined by a *cardinality constraint* on the sets and possibly some *combinatorial constraints* (e.g. matroid constraints) that rule out some subsets of $\mathcal{V}$. For simplicity, we consider only the cardinality constraint, implying that $\mathcal{C} = \{\mathcal{S} \subseteq \mathcal{V} : |\mathcal{S}| \leq K\}$, for some

[1] Note that pairwise reachability is still a stronger requirement than node semi-bandit feedback.

$K \leq n$. Since the seed nodes are the source of the diffusion, we alternatively refer to the "seed set" as the set of "source nodes".

The diffusion model $\mathcal{D}$ specifies the stochastic process under which influence is propagated across the social network once the seed set $\mathcal{S}$ is selected. Without loss of generality, we assume that all stochasticity in $\mathcal{D}$ is encoded in a random vector, referred to as the *diffusion random vector*. Note that each diffusion instance corresponds to an independent sample of the diffusion random vector from an underlying probability distribution $\mathbb{P}$ specific to the diffusion model. We denote an instantiation of the diffusion random vector by $\mathbf{w}$ and use $\mathcal{D}(\mathbf{w})$ to refer to the corresponding diffusion instance under the model $\mathcal{D}$. Note that $\mathcal{D}(\mathbf{w})$ is deterministic conditioned on $\mathbf{w}$.

The quantity $f(\mathcal{S}, p)$ refers to the *expected* number of nodes influenced by choosing the seed set $\mathcal{S}$ when the influence probabilities are given by the function p. Here, the expectation is over the possible instantiations of the random diffusion vector. Formally, $f(\mathcal{S}, p) \stackrel{\Delta}{=} \mathbb{E}_{\mathbf{w} \sim \mathbb{P}} [f(\mathcal{S}, \mathbf{w})]$ where $f(\mathcal{S}, \mathbf{w})$ is a deterministic quantity equal to the number of influenced nodes under the diffusion $\mathcal{D}(\mathbf{w})$.

We now instantiate the above framework for the independent cascade (IC) model that will be the focus of Section 2.3. For the IC model, at the beginning, all nodes in $\mathcal{S}$ are activated; subsequently, every activated node u can activate its inactive neighbour v with probability $p_{u,v}$ *once*, *independently* of the history of the process. This process continues until no activations are possible. In this case, the distribution $\mathbb{P}$ is parametrized by m influence probabilities, one for each edge and $\mathbf{w}$ is binary and is obtained by independently sampling a Bernoulli random variable $\mathbf{w}(e) \sim \text{Bern}(p(e))$ for each edge $e \in \mathcal{E}$. In this case, we use $\mathbf{w}(e) \in \{0, 1\}$ to refer to the status of edge e in the diffusion instance $\mathcal{D}(\mathbf{w})$. A diffusion instance thus corresponds to a deterministic unweighted graph; with the set of nodes $\mathcal{V}$ and the set of edges $\{e \in \mathcal{E} | \mathbf{w}(e) = 1\}$. We say that a node v is *reachable* from a node u under $\mathbf{w}$ if there is a directed path from u to v in the above deterministic graph. Note that notion of reachability in the deterministic graph and influence are equivalent for the IC model. In other words, for a given source node set $\mathcal{S} \subseteq \mathcal{V}$ and $\mathbf{w}$, we say that node $v \in \mathcal{V}$ is *influenced* if v is reachable from at least one source node in $\mathcal{S}$ under the deterministic graph induced by $\mathbf{w}$. By definition, the nodes in $\mathcal{S}$ are always influenced. Similarly, for the alternate *linear threshold model* (LT) of diffusion, $\mathbf{w}$ is also an m-dimensional vector and $\mathbb{P}$ is parametrized by the influence probabilities. For the LT model, the sum of influence probabilities corresponding to the incoming edges to any node is upper-bounded by 1. In this case, the procedure to obtain a sample $\mathbf{w}$ is as follows: for every node, choose one of its incoming edges with probability equal to the influence probability of that edge. This procedure results in the corresponding deterministic unweighted graph for the LT model.

Formally, the aim of the IM problem is to find the seed set $\mathcal{S}$ that maximizes $f(\mathcal{S}, p)$ subject to the constraint $\mathcal{S} \in \mathcal{C}$, i.e., to find $\mathcal{S}^* \in \arg\max_{\mathcal{S} \in \mathcal{C}} f(\mathcal{S}, p)$. Although IM is an NP-hard problem in general, under common diffusion models such as IC and LT, the objective function $f(\mathcal{S}, p)$ is monotone and submodular in $\mathcal{S}$, and thus, a near-optimal solution can be computed in polynomial time using the greedy algorithm (Nemhauser et al., 1978). In this paper, we refer to such an algorithm as an `ORACLE` to distinguish it from the learning algorithms discussed in following sections. Let $\mathcal{S}_G = \texttt{ORACLE}(\mathcal{G}, K, p)$ be the (possibly random) solution of an oracle `ORACLE`. For any $\alpha, \gamma \in [0, 1]$, we say that `ORACLE` is an (α, γ)-approximation oracle for a given $(\mathcal{G}, K)$ if for any p, $f(\mathcal{S}_G, p) \geq \gamma f(\mathcal{S}^*, p)$ with probability at least α. Notice that this further implies that $\mathbb{E}\left[f(\mathcal{S}_G, p)\right] \geq \alpha\gamma f(\mathcal{S}^*, p)$. We say an oracle is exact if $\alpha = \gamma = 1$. For the state of the art IM algorithm (Tang et al., 2015b), $\gamma = 1 - \frac{1}{e}$ and $\alpha = 1 - \frac{1}{m}$.

IMB is also characterized by $(\mathcal{G}, \mathcal{D}, K, p)$, but p (and possibly the diffusion model $\mathcal{D}$) is *unknown* to the agent. The agent interacts with the IMB problem for T rounds. At each round $t = 1, 2, \ldots, T$, the agent first adaptively chooses a source node set $\mathcal{S}_t \subseteq \mathcal{V}$ with cardinality K based on its prior information and past observations. Then the environment independently samples a diffusion random vector $\mathbf{w} \sim \mathbb{P}$. Note that the reward in round t is equal to the influenced nodes in the diffusion and is completely determined by $\mathcal{S}_t$ and $\mathcal{D}(\mathbf{w})$.

The agent's objective is to maximize the expected cumulative reward or equivalently minimize the cumulative regret (defined below) over the T steps. We benchmark the performance of an IMB algorithm by comparing its spread against the attainable influence assuming perfect knowledge of $\mathcal{D}$ and p. Since it is NP-hard to compute the optimal seed set even with perfect knowledge, similar to (Chen et al., 2016b), we measure the performance of an IMB algorithm by *scaled cumulative regret*. Specifically, if $\mathcal{S}_t$ is the seed set selected by the IM bandit algorithm at round t, for any $\eta \in (0, 1)$, the η-scaled cumulative regret $R^\eta(T)$ in the first T rounds is defined as

$$R^\eta(T) = \sum_{t=1}^{T} \mathbb{E}\left[R_t^\eta\right] = T \cdot f(\mathcal{S}^*) - \frac{1}{\eta}\mathbb{E}\left[\sum_{t=1}^{T} f(\mathcal{S}_t)]\right]. \tag{2.1}$$

When $\eta = 1$, $R^{\eta}(n)$ reduces to the standard expected cumulative regret $R(n)$. In our case, $\eta = \alpha\gamma$ because of the inexact oracle described above.

2.3 IM Bandits under the IC model

In this section, we focus on the IMB problem under the IC model and edge semi-bandit feedback. We describe the feedback model in Section 2.3.1 and present the algorithm and its analysis in Sections 2.3.3 and 2.3.4 respectively. We present experimental results in Section 2.3.5.

2.3.1 Feedback Model

In the edge semi-bandit feedback model, the agent observes the path along which the diffusion has travelled from the source nodes to every activated node in the network. Formally, for any edge $e = (u, v) \in \mathcal{E}$, the agent observes the realization of $\mathbf{w}_t(e)$ if and only if the start node u of the directed edge e is influenced under the realization $\mathbf{w}_t$.

2.3.2 Linear generalization

Since the number of edges in real-world social networks is large, in order to develop efficient and deployable learning algorithms, we assume that there exists a linear-generalization model for the probability weight function p. Specifically, each edge $e \in \mathcal{E}$ is associated with a *known* feature vector $x_e \in \Re^d$, where d is the dimension of the feature vector, and there is an *unknown* coefficient vector $\theta^* \in \Re^d$ such that for all $e \in \mathcal{E}$, $p(e)$ is well approximated by $\langle x_e, \theta^* \rangle$. Formally, we assume that the quantity $\max_{e \in \mathcal{E}} |p(e) - \langle x_e, \theta^* \rangle|$ is small.

Without loss of generality, we assume that $\|x_e\|_2 \leq 1$ for all $e \in \mathcal{E}$. Moreover, we use $\mathbf{X} \in \Re^{m \times d}$ to denote the feature matrix, i.e., the row of $\mathbf{X}$ associated with edge e is x_e^{T}. Note that if the agent does not have sufficient information to construct good features, it can always choose the naïve feature matrix $\mathbf{X} = \mathbf{I} \in \Re^{m \times m}$. We refer to the special case of $\mathbf{X} = \mathbf{I} \in \Re^{m \times m}$ as the *tabular* case. In the tabular case, we assume no generalization model across edges.

Algorithm 2 ICLinUCB: Independent Cascade LinUCB

```
Input: graph $\mathcal{G}$, source node set cardinality $K$, ORACLE, feature vector $x_e$'s, and algorithm parameters $\sigma, c > 0$,
Initialization: $b_0 \leftarrow 0 \in \Re^d$, $\mathbf{M}_0 \leftarrow I \in \Re^{d \times d}$
for $t = 1, 2, \ldots, T$ do
    1. set $\bar{\theta}_{t-1} \leftarrow \sigma^{-2}\mathbf{M}_{t-1}^{-1} b_{t-1}$ and the UCBs as
    $U_t(e) \leftarrow \mathrm{Proj}_{[0,1]}\left( x_e^\intercal \bar{\theta}_{t-1} + c\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \right)$ for all $e \in \mathcal{E}$
    2. choose $\mathcal{S}_t \in$ ORACLE$(\mathcal{G}, K, U_t)$, and observe the edge semi-bandit feedback
    3. update statistics:
        (a) initialize $\mathbf{M}_t \leftarrow \mathbf{M}_{t-1}$ and $b_t \leftarrow b_{t-1}$
        (b) for all observed edges $e \in \mathcal{E}$, update $\mathbf{M}_t \leftarrow \mathbf{M}_t + \sigma^{-2} x_e x_e^\intercal$ and $b_t \leftarrow b_t + x_e \mathbf{w}_t(e)$
```

2.3.3 ICLinUCB algorithm

Our proposed algorithm, Influence Maximization Linear UCB (ICLinUCB), is detailed in Algorithm 2. Notice that ICLinUCB represents its past observations as a positive-definite matrix (*Gram matrix*) $\mathbf{M}_t \in \Re^{d \times d}$ and a vector $b_t \in \Re^d$. Specifically, let $\mathbf{X}_t$ be a matrix whose rows are the feature vectors of all the observed edges in the t preceding rounds. Let Y_t be a binary column vector encoding the realizations of these observed edges in the t rounds. Then $\mathbf{M}_t = \mathbf{I} + \sigma^{-2}\mathbf{X}_t^\intercal \mathbf{X}_t$ and $b_t = \mathbf{X}_t^\intercal Y_t$. Here σ is the standard deviation of the noise in the observations.

At each round t, ICLinUCB operates in three steps: First, it computes an upper confidence bound $U_t(e)$ for each edge $e \in \mathcal{E}$. Note that $\mathrm{Proj}_{[0,1]}(\cdot)$ projects a real number into interval $[0, 1]$ to ensure that it is a probability. Second, it chooses a set of source nodes based on the given ORACLE with U_t as its input set of probabilities. Finally, it receives the edge semi-bandit feedback and uses it to update $\mathbf{M}_t$ and b_t.

Note that ICLinUCB is computationally efficient as long as ORACLE is computationally efficient. Specifically, at each round t, the computational complexities of both Step 1 and 3 of ICLinUCB are $\mathcal{O}\left(md^2\right)$. Notice that in a practical implementation, we store $\mathbf{M}_t^{-1}$ instead of $\mathbf{M}_t$. Moreover, $\mathbf{M}_t \leftarrow \mathbf{M}_t + \sigma^{-2} x_e x_e^\intercal$ is equivalent to $\mathbf{M}_t^{-1} \leftarrow \mathbf{M}_t^{-1} - \frac{\mathbf{M}_t^{-1} x_e x_e^\intercal \mathbf{M}_t^{-1}}{x_e^\intercal \mathbf{M}_t^{-1} x_e + \sigma^2}$ by the Sherman-Morrison formula.

It is worth pointing out that in the tabular case, ICLinUCB reduces to the CUCB algorithm in the previous work(Chen et al., 2013b), in the sense that the confidence radii in ICLinUCB are the same as those in CUCB, up to logarithmic factors. That is, CUCB can be viewed as a special case of ICLinUCB with $\mathbf{X} = \mathbf{I}$.

2.3.4 Analysis

In this section, we give a regret bound for ICLinUCB for the case when $p(e) = x_e^\mathsf{T} \theta^*$ for all $e \in \mathcal{E}$, i.e., the linear generalization is perfect. Our main contribution is a regret bound that scales with a new complexity metric, *maximum observed relevance*, which depends on *both* the topology of $\mathcal{G}$ and the probability weight function p. We highlight this as most known results for this problem are worst case, and some of them do not depend on probability weight function at all.

Complexity metric: Maximum observed relevance

We start by defining some terminology. For given directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and source node set $\mathcal{S} \subseteq \mathcal{V}$, we say an edge $e \in \mathcal{E}$ is *relevant* to a node $v \in \mathcal{V} \setminus \mathcal{S}$ under $\mathcal{S}$ if there exists a path p from a source node $s \in \mathcal{S}$ to v such that (1) $e \in p$ and (2) p does not contain another source node other than s. Notice that with a given $\mathcal{S}$, whether or not a node $v \in \mathcal{V} \setminus \mathcal{S}$ is influenced only depends on the binary weights $\mathbf{w}$ on its relevant edges. For any edge $e \in \mathcal{E}$, we define $N_{\mathcal{S},e}$ as the number of nodes in $\mathcal{V} \setminus \mathcal{S}$ it is relevant to, and define $P_{\mathcal{S},e}$ as the conditional probability that e is observed given $\mathcal{S}$,

$$N_{\mathcal{S},e} \stackrel{\Delta}{=} \textstyle\sum_{v \in \mathcal{V} \setminus \mathcal{S}} \mathbf{1}\left\{e \text{ is relevant to } v \text{ under } \mathcal{S}\right\} \quad \text{and} \quad P_{\mathcal{S},e} \stackrel{\Delta}{=} \mathbb{P}\left(e \text{ is observed} \,|\, \mathcal{S}\right). \tag{2.2}$$

Notice that $N_{\mathcal{S},e}$ only depends on the topology of $\mathcal{G}$, while $P_{\mathcal{S},e}$ depends on *both* the topology of $\mathcal{G}$ and the probability weight function p. The *maximum observed relevance* C_* is defined as the maximum (over $\mathcal{S}$) square of $N_{\mathcal{S},e}$'s weighted by $P_{\mathcal{S},e}$'s,

$$C_* \stackrel{\Delta}{=} \max\nolimits_{\mathcal{S}\colon |\mathcal{S}|=K} \sqrt{\textstyle\sum_{e \in \mathcal{E}} N_{\mathcal{S},e}^2 P_{\mathcal{S},e}}. \tag{2.3}$$

Note that C_* also depends on both the topology of $\mathcal{G}$ and the probabilities p. However, C_* can be bounded from above only based on the topology of $\mathcal{G}$ or the size of the problem, i.e., $n = |\mathcal{V}|$ and m. Specifically, by defining $C_\mathcal{G} \stackrel{\Delta}{=} \max_{\mathcal{S}\colon |\mathcal{S}|=K} \sqrt{\sum_{e \in \mathcal{E}} N_{\mathcal{S},e}^2}$, we have

$$C_* \leq C_\mathcal{G} = \max\nolimits_{\mathcal{S}\colon |\mathcal{S}|=K} \sqrt{\textstyle\sum_{e \in \mathcal{E}} N_{\mathcal{S},e}^2} \leq (n-K)\sqrt{m} = \mathcal{O}\left(n\sqrt{m}\right) = \mathcal{O}\left(n^2\right), \tag{2.4}$$

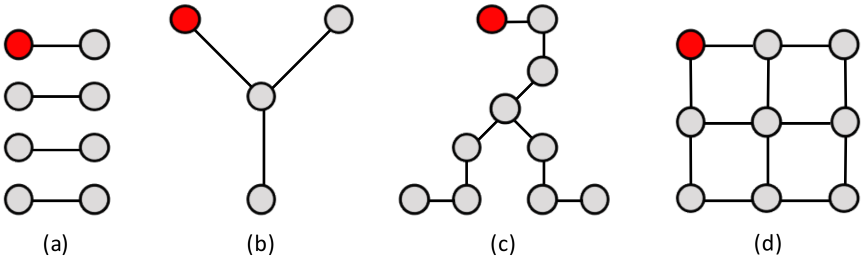


Figure 2.1: **a**. Bar graph on 8 nodes. **b**. Star graph on 4 nodes. **c**. Ray graph on 10 nodes. **d**. Grid graph on 9 nodes. Each undirected edge denotes two directed edges in opposite directions.

topology	$C_{\mathcal{G}}$ (worst-case C_*)	$R^{\alpha\gamma}(T)$ for general $\mathbf{X}$	$R^{\alpha\gamma}(T)$ for $\mathbf{X} = \mathbf{I}$
bar graph	$\mathcal{O}(\sqrt{K})$	$\widetilde{\mathcal{O}}\left(dK\sqrt{T}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n\sqrt{KT}/(\alpha\gamma)\right)$
star graph	$\mathcal{O}(n\sqrt{K})$	$\widetilde{\mathcal{O}}\left(dn^{\frac{3}{2}}\sqrt{KT}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n^2\sqrt{KT}/(\alpha\gamma)\right)$
ray graph	$\mathcal{O}(n^{\frac{5}{4}}\sqrt{K})$	$\widetilde{\mathcal{O}}\left(dn^{\frac{7}{4}}\sqrt{KT}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n^{\frac{9}{4}}\sqrt{KT}/(\alpha\gamma)\right)$
tree graph	$\mathcal{O}(n^{\frac{3}{2}})$	$\widetilde{\mathcal{O}}\left(dn^2\sqrt{T}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n^{\frac{5}{2}}\sqrt{T}/(\alpha\gamma)\right)$
grid graph	$\mathcal{O}(n^{\frac{3}{2}})$	$\widetilde{\mathcal{O}}\left(dn^2\sqrt{T}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n^{\frac{5}{2}}\sqrt{T}/(\alpha\gamma)\right)$
complete graph	$\mathcal{O}(n^2)$	$\widetilde{\mathcal{O}}\left(dn^3\sqrt{T}/(\alpha\gamma)\right)$	$\widetilde{\mathcal{O}}\left(n^4\sqrt{T}/(\alpha\gamma)\right)$

Table 2.1: $C_{\mathcal{G}}$ and *worst-case* regret bounds for different graph topologies.

where $C_{\mathcal{G}}$ is the maximum/worst-case (over p) C_* for the directed graph $\mathcal{G}$, and the maximum is obtained by setting $p(e) = 1$ for all $e \in \mathcal{E}$. Since $C_{\mathcal{G}}$ is worst-case, it might be very far away from C_* if the influence probabilities are small. Indeed, this is what we expect in typical real-world situations (Goyal et al., 2010). Notice also that if $\max_{e\in\mathcal{E}} p(e) \to 0$, then $P_{\mathcal{S},e} \to 0$ for all $e \notin \mathcal{E}(\mathcal{S})$ and $P_{\mathcal{S},e} = 1$ for all $e \in \mathcal{E}(\mathcal{S})$, where $\mathcal{E}(\mathcal{S})$ is the set of edges with start node in $\mathcal{S}$, hence we have $C_* \to C^0_{\mathcal{G}} \overset{\Delta}{=} \max_{\mathcal{S}:\, |\mathcal{S}|=K} \sqrt{\sum_{e\in\mathcal{E}(\mathcal{S})} N^2_{\mathcal{S},e}}$. In particular, if K is small, $C^0_{\mathcal{G}}$ is much less than $C_{\mathcal{G}}$ in many topologies. For example, in a complete graph with $K = 1$, $C_{\mathcal{G}} = \Theta(n^2)$ while $C^0_{\mathcal{G}} = \Theta(n^{\frac{3}{2}})$. Finally, it is worth pointing out that there exist situations $(\mathcal{G}, p)$ such that $C_* = \Theta(n^2)$. One such example is when $\mathcal{G}$ is a complete graph with n nodes and $p(e) = n/(n+1)$ for all edges e in this graph.

To give more intuition about this quantity, we illustrate how $C_{\mathcal{G}}$, the *worst-case* C_*, varies with four graph topologies in Figure 2.1: bar, star, ray, and grid, as well as two other topologies: general tree and complete graph. The bar graph (Figure 2.1a) is a graph where nodes i and $i+1$ are connected when i is odd. The star graph (Figure 2.1b) is a graph where node 1 is central and all remaining nodes $i \in \mathcal{V} \setminus \{1\}$ are connected to it. The distance between any two of these nodes is 2. The ray graph (Figure 2.1c) is a star graph with $k = \lceil \sqrt{n-1} \rceil$ arms, where node 1 is central and each arm contains either $\lceil (n-1)/k \rceil$ or $\lfloor (n-1)/k \rfloor$ nodes connected in a line. The distance between any two nodes in this graph is $\mathcal{O}(\sqrt{n})$. The grid graph (Figure 2.1d) is a classical non-tree graph with $\mathcal{O}(n)$ edges.

To see how $C_{\mathcal{G}}$ varies with the graph topology, we start with the simplified case when $K = |\mathcal{S}| = 1$. In the bar graph (Figure 2.1a), only one edge is relevant to a node $v \in \mathcal{V} \setminus \mathcal{S}$ and all the other edges are not relevant to any nodes. Therefore, $C_{\mathcal{G}} \leq 1$. In the star graph (Figure 2.1b), for any s, at most one edge is relevant to at most $n-1$ nodes and the remaining edges are relevant to at most one node. In this case, $C_{\mathcal{G}} \leq \sqrt{n^2+n} = \mathcal{O}(n)$. In the ray graph (Figure 2.1c), for any s, at most $\mathcal{O}(\sqrt{n})$ edges are relevant to $n-1$ nodes and the remaining edges are relevant to at most $\mathcal{O}(\sqrt{n})$ nodes. In this case, $C_{\mathcal{G}} = \mathcal{O}(\sqrt{n^{\frac{1}{2}} n^2 + nn}) = \mathcal{O}(n^{\frac{5}{4}})$. Finally, recall that for all graphs we can bound $C_{\mathcal{G}}$ by $\mathcal{O}(n\sqrt{m})$, regardless of K. Hence, for the grid graph (Figure 2.1d) and general tree graph, $C_{\mathcal{G}} = \mathcal{O}(n^{\frac{3}{2}})$ since $m = \mathcal{O}(n)$; for the complete graph $C_{\mathcal{G}} = \mathcal{O}(n^2)$ since $m = \mathcal{O}(n^2)$. Clearly, $C_{\mathcal{G}}$ varies widely with the topology of the graph. The second column of Table 2.1 summarizes how $C_{\mathcal{G}}$ varies with the above-mentioned graph topologies for general $K = |\mathcal{S}|$.

Regret guarantees

We obtain the following regret guarantees for `ICLinUCB` in terms of the complexity metric C_*.

Theorem 1. *Assume that (1) $p(e) = x_e^{\mathsf{T}} \theta^*$ for all $e \in \mathcal{E}$ and (2)* `ORACLE` *is an (α, γ)-approximation algorithm. Let D be a known upper bound on $\|\theta^*\|_2$, if we apply* `ICLinUCB` *with $\sigma = 1$ and*

$$c = \sqrt{d \log\left(1 + \frac{Tm}{d}\right) + 2\log\left(T(n+1-K)\right)} + D, \tag{2.5}$$

then we have

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma}\sqrt{dTm\log_2\left(1+\frac{Tm}{d}\right)}+1 = \widetilde{\mathcal{O}}\left(dC_*\sqrt{mT}/(\alpha\gamma)\right) \tag{2.6}$$

$$\leq \widetilde{\mathcal{O}}\left(d(n-K)m\sqrt{T}/(\alpha\gamma)\right). \tag{2.7}$$

Moreover, if the feature matrix $\mathbf{X} = \mathbf{I} \in \Re^{m\times m}$ *(i.e., the tabular case), we have*

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma}\sqrt{Tm\log_2(1+T)}+1 = \widetilde{\mathcal{O}}\left(mC_*\sqrt{T}/(\alpha\gamma)\right) \tag{2.8}$$

$$\leq \widetilde{\mathcal{O}}\left((n-K)m^{\frac{3}{2}}\sqrt{T}/(\alpha\gamma)\right). \tag{2.9}$$

Please refer to Appendix A.1 for the proof of Theorem 1, that we outline below. We now briefly comment on the regret bounds in Theorem 1.

Topology-dependent bounds: Since C_* is topology-dependent, the regret bounds in Equations 2.6 and 2.8 are also topology-dependent. Table 2.1 summarizes the regret bounds for each topology[2] discussed above. Since the regret bounds in Table 2.1 are the worst-case regret bounds for a given topology, more general topologies have larger regret bounds. For instance, the regret bounds for a tree are larger than their counterparts for star and ray, since star and ray are special cases of a tree. The grid and tree can also be viewed as special cases of complete graphs by setting $p(e) = 0$ for some $e \in \mathcal{E}$, hence complete graph has larger regret bounds. As explained earlier, in practice we expect C_* to be far smaller due to influence probabilities.

Tighter bounds in tabular case and under exact oracle: Notice that for the tabular case with feature matrix $\mathbf{X} = \mathbf{I}$ and $d = m$, $\widetilde{\mathcal{O}}(\sqrt{m})$ tighter regret bounds are obtained in Equations 2.8 and 2.9. Also notice that the $\widetilde{\mathcal{O}}(1/(\alpha\gamma))$ factor is due to the fact that `ORACLE` is an (α, γ)-approximation oracle. If `ORACLE` solves the IM problem exactly (i.e., $\alpha = \gamma = 1$), then $R^{\alpha\gamma}(T) = R(T)$.

Tightness of our regret bounds: First, note that our regret bound in the bar case with $K = 1$ matches the regret bound of the classic `LinUCB` algorithm. Specifically, with perfect linear generalization, this case is equivalent to a linear bandit problem with n arms and feature dimension d. From Table 2.1, our regret bound in this case is $\widetilde{\mathcal{O}}\left(d\sqrt{T}\right)$,

[2] The regret bound for bar graph is based on Theorem 8 in the appendix, which is a stronger version of Theorem 1 for disconnected graph.

which matches the known regret bound of `LinUCB` that can be obtained by the technique of (Abbasi-Yadkori et al., 2011). Second, we briefly discuss the tightness of the regret bound in Equation 2.7 for a general graph with n nodes and m edges. Note that the $\widetilde{\mathcal{O}}(\sqrt{T})$-dependence on time is near-optimal, and the $\widetilde{\mathcal{O}}(d)$-dependence on feature dimension is standard in linear bandits (Abbasi-Yadkori et al., 2011; Wen et al., 2015a), since $\widetilde{\mathcal{O}}(\sqrt{d})$ results are only known for impractical algorithms. The $\widetilde{\mathcal{O}}(n-K)$ factor is due to the fact that the reward in this problem is from K to n, rather than from 0 to 1. To explain the $\widetilde{\mathcal{O}}(m)$ factor in this bound, notice that one $\widetilde{\mathcal{O}}(\sqrt{m})$ factor is due to the fact that at most $\widetilde{\mathcal{O}}(m)$ edges might be observed at each round (see Theorem 1), and is intrinsic to the problem similar to combinatorial semi-bandits (Kveton et al., 2015c); another $\widetilde{\mathcal{O}}(\sqrt{m})$ factor is due to linear generalization (see Lemma 7) and might be removed by better analysis. We conjecture that our $\widetilde{\mathcal{O}}\left(d(n-K)m\sqrt{T}/(\alpha\gamma)\right)$ regret bound in this case is at most $\widetilde{\mathcal{O}}(\sqrt{md})$ away from being tight.

Proof sketch

We now outline the proof of Theorem 1. For each round $t \leq T$, we define the favourable event $\xi_{t-1} = \{|x_e^\mathsf{T}(\overline{\theta}_{\tau-1} - \theta^*)| \leq c\sqrt{x_e^\mathsf{T}\mathbf{M}_{\tau-1}^{-1}x_e},\ \forall e \in \mathcal{E},\ \forall \tau \leq t\}$, and the unfavourable event $\overline{\xi}_{t-1}$ as the complement of ξ_{t-1}. If we decompose $\mathbb{E}[R_t^{\alpha\gamma}]$, the $(\alpha\gamma)$-scaled expected regret at round t, over events ξ_{t-1} and $\overline{\xi}_{t-1}$, and bound $R_t^{\alpha\gamma}$ on event $\overline{\xi}_{t-1}$ using the naïve bound $R_t^{\alpha\gamma} \leq n - K$, then,

$$\mathbb{E}[R_t^{\alpha\gamma}] \leq \mathbb{P}\left(\xi_{t-1}\right)\mathbb{E}\left[R_t^{\alpha\gamma}|\xi_{t-1}\right] + \mathbb{P}\left(\overline{\xi}_{t-1}\right)[n-K].$$

By choosing c as specified by Equation 2.5, we have $\mathbb{P}\left(\overline{\xi}_{t-1}\right)[n-K] < 1/T$ (see Lemma 3 in the appendix). On the other hand, notice that by definition of ξ_{t-1}, $p(e) \leq U_t(e)$, $\forall e \in \mathcal{E}$ under event ξ_{t-1}. Using the monotonicity of the spread f in the probabilities, and the fact that `ORACLE` is an (α, γ)-approximation algorithm, we have

$$\mathbb{E}\left[R_t^{\alpha\gamma}|\xi_{t-1}\right] \leq \mathbb{E}\left[f(\mathcal{S}_t, U_t) - f(\mathcal{S}_t, p)|\xi_{t-1}\right]/(\alpha\gamma).$$

The next observation is that, from the linearity of expectation, the gap $f(\mathcal{S}_t, U_t) - f(\mathcal{S}_t, p)$ decomposes over nodes $v \in \mathcal{V} \setminus \mathcal{S}_t$. Specifically, for any source node set $\mathcal{S} \subseteq \mathcal{V}$, any probability weight function $p : \mathcal{E} \to [0,1]$, and any node $v \in \mathcal{V}$, we define $f(\mathcal{S}, p, v)$ as the probability that node v is influenced if the source node set is $\mathcal{S}$ and the probability weight function is p. Hence, we have

$$f(\mathcal{S}_t, U_t) - f(\mathcal{S}_t, p) = \textstyle\sum_{v \in \mathcal{V} \setminus \mathcal{S}_t} \left[f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v) \right].$$

In the appendix, we show that under any weight function, the diffusion process from the source node set S_t to the target node v can be modeled as a Markov chain. Hence, weight function U_t and p give us two Markov chains with the same state space but different transition probabilities. $f(S_t, U_t, v) - f(S_t, p, v)$ can be recursively bounded based on the state diagram of the Markov chain under weight function p. With some algebra, Theorem 9 in Appendix A.1 bounds $f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v)$ by the edge-level gap $U_t(e) - p(e)$ on the observed relevant edges for node v,

$$f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v) \leq \textstyle\sum_{e \in \mathcal{E}_{\mathcal{S}_t, v}} \mathbb{E}\left[\mathbf{1}\left\{ O_t(e) \right\} \left[U_t(e) - p(e) \right] | \mathcal{H}_{t-1}, \mathcal{S}_t \right], \tag{2.10}$$

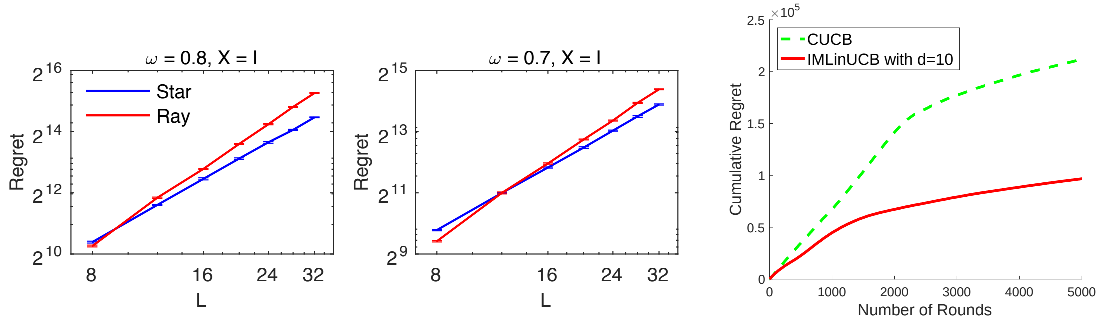
for any t, any set of past observations and $\mathcal{S}_t$ such that ξ_{t-1} holds, and any $v \in \mathcal{V} \setminus \mathcal{S}_t$, where $\mathcal{E}_{\mathcal{S}_t, v}$ is the set of edges relevant to v and $O_t(e)$ is the event that edge e is observed at round t. Based on Equation 2.10, we can prove Theorem 1 using the standard linear-bandit techniques (see Appendix A.1).

2.3.5 Experiments

In this section, we present a synthetic experiment in order to empirically validate our upper bounds on the regret. Next, we evaluate our algorithm on a real-world Facebook subgraph.

Stars and rays

In the first experiment, we evaluate `ICLinUCB` on undirected stars and rays (Figure 2.1) and validate that the regret grows with the number of nodes n and the maximum observed relevance C_* as shown in Table 2.1. We focus on the tabular case ($\mathbf{X} = \mathbf{I}$) with $K = |\mathcal{S}| = 1$, where the IM problem can be solved exactly. We vary the number of nodes n; and edge weight $p(e) = \omega$, which is the same for all edges e. We run `ICLinUCB` for $T = 10^4$ steps and verify that it converges to the optimal solution in each experiment. We report the T-step regret of `ICLinUCB` for $8 \leq n \leq 32$ in Figure 2.2a. Recall that from Table 2.1, $R(T) = \widetilde{\mathcal{O}}(n^2)$ for star and $R(T) = \widetilde{\mathcal{O}}(n^{\frac{9}{4}})$ for ray.



(a) Stars and rays: The log-log plots of the T-step regret of `ICLinUCB` in two graph topologies after $T = 10^4$ steps. We vary the number of nodes n and the mean edge weight ω.

(b) Subgraph of the Facebook network

Figure 2.2: Experimental results for (a) Representative graph topologies (b) Subgraph of the Facebook network. The regret for ray and star graphs scales as suggested by our theoretical regret bounds. For the Facebook subgraph, we observe that the linear generalization across edges results in lower cumulative regret as compared to CUCB that considers each edge independently.

We numerically estimate the growth of regret in n, the exponent of n, in the log-log space of n and regret. In particular, since $\log(f(n)) = p\log(n) + \log(c)$ for any $f(n) = cn^p$ and $c > 0$, both p and $\log(c)$ can be estimated by linear regression in the new space. For star graphs with $\omega = 0.8$ and $\omega = 0.7$, our estimated growths are respectively $\mathcal{O}(n^{2.040})$ and $\mathcal{O}(n^{2.056})$, which are close to the expected $\widetilde{\mathcal{O}}(n^2)$. For ray graphs with $\omega = 0.8$ and $\omega = 0.7$, our estimated growth are respectively $\mathcal{O}(n^{2.488})$ and $\mathcal{O}(n^{2.467})$, which are again close to the expected $\widetilde{\mathcal{O}}(n^{\frac{9}{4}})$. This shows that maximum observed relevance C_* is a reasonable complexity metric for these two topologies.

Subgraph of the Facebook network

In the second experiment, we demonstrate the potential performance gain of `ICLinUCB` in real-world influence maximization semi-bandit problems by exploiting linear generalization across edges. Specifically, we compare `ICLinUCB` with `CUCB` in a subgraph of the Facebook network from (Leskovec and Krevl, 2014). The subgraph has $n = |\mathcal{V}| = 327$ nodes and $m = |\mathcal{E}| = 5038$ directed edges. Since the true probability weight function p is not available, we independently sample $p(e)$'s from the uniform distribution $U(0, 0.1)$ and treat them as ground-truth. Note that this range of probabilities is guided by empirical evidence in (Goyal et al., 2010; Barbieri et al., 2013). We set $T = 5000$ and $K = 10$ in this experiment. For `ICLinUCB`, we choose $d = 10$ and generate edge feature x_e's as follows: we first use the `node2vec` algorithm (Grover and Leskovec, 2016) to generate a node feature in $\Re^d$ for each node $v \in \mathcal{V}$; then for each edge e, we generate x_e as the element-wise product of node features of the two nodes connected to e. Note that the linear generalization in this experiment is imperfect in the sense that $\min_{\theta \in \Re^d} \max_{e \in \mathcal{E}} |p(e) - x_e^T \theta| > 0$. For both `CUCB` and `ICLinUCB`, we choose `ORACLE` as the state-of-the-art offline IM algorithm proposed in (Tang et al., 2014). To compute the cumulative regret, we compare against a fixed seed set $\mathcal{S}^*$ obtained by using the true set of probabilities as input to the oracle proposed in (Tang et al., 2014). We average the empirical cumulative regret over 10 independent runs, and plot the results in Figure 2.2b. The experimental results show that compared with `CUCB`, `ICLinUCB` can significantly reduce the cumulative regret by exploiting linear generalization across $p(e)$'s. This shows that the influence probability for an edge depends on its local graph neighbourhood and that we can use graph representation learning techniques to exploit this underlying structure in order to learn more efficiently.

2.4 Model-Independent IM Bandits

In the previous section, we tackled the IMB problem under the specific IC model of diffusion. In practical scenarios, it is not clear how to choose a "good" model of diffusion that explains the observed data. In this section, we develop a model-agnostic solution to the IMB problem. We first propose a model-independent parametrization and the corresponding surrogate objective in Section 2.4.1. In Section 2.4.2, we describe the feedback model. We present the algorithm and its analysis in Sections 2.4.4 and 2.4.5 respectively. We describe the implementation details in Section 2.4.6 and present the experimental results in Section 2.4.7.

2.4.1 Surrogate Objective

Let us first define some useful notation: we use the indicator $\mathbb{1}\big(\mathcal{S}, v, \mathcal{D}(\mathbf{w})\big) \in \{0, 1\}$ to denote whether or not the node v is influenced under the seed set $\mathcal{S}$ and the particular realization $\mathcal{D}(\mathbf{w})$. For a given $(\mathcal{G}, \mathcal{D})$, once the seed set $\mathcal{S} \subseteq \mathcal{C}$ is chosen, for each $v \in \mathcal{V}$, we use $f(\mathcal{S}, v)$ to denote the probability that node v is influenced under the seed set $\mathcal{S}$,

$$f(\mathcal{S}, v) = \mathbb{E}_{\mathbf{w}}\left[\mathbb{1}\big(\mathcal{S}, v, \mathcal{D}(\mathbf{w})\big)\big|\mathcal{S}\right] \tag{2.11}$$

Note that the expected spread is given as: $f(\mathcal{S}) = \sum_{v \in \mathcal{V}} f(\mathcal{S}, v)$ and recall that the set $\mathcal{S}^* \subseteq \mathcal{C}$ maximizes it.

We now introduce the notion of *pairwise reachability*: for every pair of nodes $u, v \in \mathcal{V}$, we define the pairwise reachability $q_{u,v}$ from u to v as the probability that v will be influenced, if u is the only seed node under graph $\mathcal{G}$ and diffusion model $\mathcal{D}$, implying that $q_{u,v} = f(\{u\}, v)$. Given q, we define the *maximal pairwise reachability* from the source set $\mathcal{S}$ to the target node v as follows:

$$\widetilde{f}(\mathcal{S}, v, q) = \max\nolimits_{u \in \mathcal{S}} \; q_{u,v} \tag{2.12}$$

We define the surrogate objective function in terms of these maximal pairwise reachabilities as follows:

$$\widetilde{f}(\mathcal{S}, q) = \sum_{v \in \mathcal{V}} \widetilde{f}(\mathcal{S}, v, q) \tag{2.13}$$

Let $\widetilde{\mathcal{S}}$ be the solution to the following problem:

$$\widetilde{\mathcal{S}} \in \arg\max\nolimits_{\mathcal{S} \in \mathcal{C}} \widetilde{f}(\mathcal{S}, q) \tag{2.14}$$

Note that for all q and irrespective of the diffusion model $\mathcal{D}$, $\widetilde{f}(\mathcal{S}, q)$ is always monotone and submodular in $\mathcal{S}$ (Krause and Golovin, 2012) and can be maximized using the greedy algorithm in (Nemhauser et al., 1978).

To quantify the quality of the surrogate, we assume that $\mathcal{D}$ is any diffusion model satisfying the following monotonicity assumption:

Assumption 1. *The spread $f(\mathcal{S}, v)$ is monotone in $\mathcal{S}$, implying that for any $v \in \mathcal{V}$ and any subsets $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{V}$, $f(\mathcal{S}_1, v) \leq f(\mathcal{S}_2, v)$.*

Note that all progressive diffusion models (Kempe et al., 2003; Gomez Rodriguez et al., 2012; Li et al., 2013) where an influenced user can not become inactive again satisfy Assumption 1.

We define the *surrogate approximation factor* as $\rho = \widetilde{f}(\widetilde{\mathcal{S}}, q)/f(\mathcal{S}^*)$. The next theorem, (proved in Appendix A.4) obtains the following upper and lower bounds on ρ:

Theorem 2. *For any graph $\mathcal{G}$, seed set $\mathcal{S} \in \mathcal{C}$, and diffusion model $\mathcal{D}$ satisfying Assumption 1,*

1 $\widetilde{f}(\mathcal{S}, q) \leq f(\mathcal{S})$,

2 Furthermore, if $f(\mathcal{S})$ is submodular in $\mathcal{S}$, then $1/K \leq \rho \leq 1$.

The above theorem implies that for any diffusion model satisfying Assumption 1, maximizing $\widetilde{f}(\mathcal{S}, q)$ is equivalent to maximizing a lower-bound on the true spread $f(\mathcal{S})$. For the common independent cascade and linear threshold models, $f(\mathcal{S})$ is both monotone and submodular in $\mathcal{S}$, and the approximation factor can be no worse than $1/K$. In Section 2.4.7, we observe that in cases of practical interest, $\widetilde{f}(\mathcal{S}, q)$ is a good approximation to $f(\mathcal{S})$ and that ρ is typically much larger than $1/K$.

In this section, we use `ORACLE` to refer to the algorithm for solving the maximization problem in Equation 2.14. Let $\widehat{\mathcal{S}} \stackrel{\Delta}{=} \texttt{ORACLE}(\mathcal{G}, \mathcal{C}, p)$ be the seed set output by the oracle. For any $\alpha \in [0, 1]$, we say that `ORACLE` is an α-approximation algorithm if for all $q : \mathcal{V} \times \mathcal{V} \to [0, 1]$, $\widetilde{f}(\widehat{\mathcal{S}}, q) \geq \alpha \widetilde{f}(\widetilde{\mathcal{S}}, q)$. For our particular case, since $\widetilde{f}(\mathcal{S}, q)$ is submodular, the greedy algorithm gives an $\alpha = 1 - 1/e$ approximation (Nemhauser et al., 1978). Hence, given the knowledge of q, we can obtain a $\rho\alpha$-approximate solution to the IM problem without knowledge of the underlying diffusion model $\mathcal{D}$.

2.4.2 Feedback Model

We now describe the IM semi-bandit feedback model referred to as *pairwise influence feedback*. Under this feedback model, at the end of each round t, the agent observes the quantity $\mathbb{1}\big(\{u\}, v, \mathcal{D}(\mathbf{w}_t)\big)$ for all $u \in \mathcal{S}_t$ and all $v \in \mathcal{V}$. In other words, they observe whether or not node v would have been influenced, if the agent had selected $\{u\}$ as the seed set under the diffusion instance $\mathcal{D}(\mathbf{w}_t)$. Note that this assumption is strictly weaker than (and is implied by) the edge semi-bandit feedback model in the previous section: from edge semi-bandit feedback, we can identify the edges along which the diffusion travelled, and thus, determine whether a particular source node is responsible for activating a target node. However, from pairwise feedback, it is impossible to infer a unique edge level feedback.

2.4.3 Linear Generalization

The proposed parametrization in terms of reachability probabilities results in $O(n^2)$ parameters that need to be learned. Without any additional assumptions, this becomes intractable for large networks. To develop statistically efficient algorithms for large-scale IM semi-bandits, we make a linear generalization assumption similar to the previous section. Specifically, we assume that each node $v \in \mathcal{V}$ is associated with two vectors of dimension d, the seed (source) weight vector $\theta^*_v \in \Re^d$ and the target feature vector $\mathbf{x}_v \in \Re^d$. We assume that the target feature $\mathbf{x}_v$ is known, whereas θ^*_v is unknown and needs to be learned. The linear generalization assumption is stated as:

Assumption 2. *For all $u, v \in \mathcal{V}$, $q_{u,v}$ can be well approximated by the inner product of θ^*_u and $\mathbf{x}_v$, i.e.,*

$$q_{u,v} = \langle \theta^*_u, \mathbf{x}_v \rangle \triangleq \mathbf{x}_v^\top \theta^*_u$$

Note that for the *tabular case* (the case without generalization across $q_{u,v}$), we can always choose $\mathbf{x}_v = e_v \in \Re^n$ and $\theta^*_u = [q_{u,1}, \ldots, q_{u,n}]^T$, where e_v is an n-dimensional indicator vector with the v-th element equal to 1 and all other elements equal to 0. We discuss an approach to construct features based on the unweighted graph Laplacian in Section 2.4.6. We use the matrix $X \in \Re^{d \times n}$ to encode the target features. Specifically, for $v = 1, \ldots, n$, the v-th column of X is set as $\mathbf{x}_v$. Note that $X = I \in \Re^{n \times n}$ in the tabular case.

Finally, note that under Assumption 2, estimating the reachability probabilities becomes equivalent to estimating n (one for each source) d-dimensional weight vectors. This implies that Assumption 2 reduces the number of parameters to learn from $O(n^2)$ to $O(dn)$, and thus, is important for developing statistically efficient algorithms for large-scale problems.

2.4.4 Algorithm

In this section, we propose a LinUCB-based IM semi-bandit algorithm, called *diffusion-independent LinUCB* (`DILinUCB`), whose pseudocode is in Algorithm 3. As its name suggests, `DILinUCB` is applicable to IM semi-bandits with any diffusion model $\mathcal{D}$ satisfying Assumption 1. The only requirement to apply `DILinUCB` is that the IM semi-bandit provides the pairwise influence feedback described earlier.

Algorithm 3 Diffusion-Independent LinUCB (DILinUCB)

```
1:  **Input:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{C}$, oracle ORACLE, target feature matrix $X \in \mathbb{R}^{d \times n}$, algorithm parameters $c, \lambda, \sigma > 0$
2:  Initialize $\Sigma_{u,0} \leftarrow \lambda I_d$, $\mathbf{b}_{u,0} \leftarrow \mathbf{0}$, $\widehat{\theta}_{u,0} \leftarrow \mathbf{0}$ for all $u \in \mathcal{V}$, and UCB $\overline{q}_{u,v} \leftarrow 1$ for all $u, v \in \mathcal{V}$
3:  **for** $t = 1$ **to** $T$ **do**
4:      Choose $\mathcal{S}_t \leftarrow$ ORACLE$(\mathcal{G}, \mathcal{C}, \overline{q})$
5:      **for** $u \in \mathcal{S}_t$ **do**
6:          Get pairwise influence feedback $\boldsymbol{y}_{u,t}$
7:          $\mathbf{b}_{u,t} \leftarrow \mathbf{b}_{u,t-1} + X\boldsymbol{y}_{u,t}$
8:          $\Sigma_{u,t} \leftarrow \Sigma_{u,t-1} + \sigma^{-2} X X^T$
9:          $\widehat{\theta}_{u,t} \leftarrow \sigma^{-2} \Sigma_{u,t}^{-1} \mathbf{b}_{u,t}$
10:         $\overline{q}_{u,v} \leftarrow \text{Proj}_{[0,1]} \left[ \langle \widehat{\theta}_{u,t}, \mathbf{x}_v \rangle + c \|\mathbf{x}_v\|_{\Sigma_{u,t}^{-1}} \right]$, $\forall v \in \mathcal{V}$
11:     **for** $u \notin \mathcal{S}_t$ **do**
12:         $\mathbf{b}_{u,t} = \mathbf{b}_{u,t-1}$
13:         $\Sigma_{u,t} = \Sigma_{u,t-1}$
```

The inputs to DILinUCB include the network topology $\mathcal{G}$, the collection of the feasible sets $\mathcal{C}$, the optimization algorithm ORACLE, the target feature matrix X, and three algorithm parameters $c, \lambda, \sigma > 0$. The parameter λ is a regularization parameter, whereas σ is proportional to the noise in the observations and hence controls the learning rate. For each source node $u \in \mathcal{V}$ and round t, we define the Gram matrix $\Sigma_{u,t} \in \Re^{d \times d}$, and $\mathbf{b}_{u,t} \in \Re^d$ as the vector summarizing the past propagations from u. The vector $\widehat{\theta}_{u,t}$ is the source weight estimate for node u at round t. The mean reachability probability from u to v is given by $\langle \widehat{\theta}_{u,t}, \mathbf{x}_v \rangle$, whereas its variance is given as $\|\mathbf{x}_v\|_{\Sigma_{u,t}^{-1}} = \sqrt{\mathbf{x}_v^T \Sigma_{u,t}^{-1} \mathbf{x}_v}$. Note that Σ_u and $\mathbf{b}_u$ are sufficient statistics for computing UCB estimates $\overline{q}_{u,v}$ for all $v \in \mathcal{V}$. The parameter c trades off the mean and the standard deviation in the UCB estimates and thus controls the "degree of optimism" of the algorithm.

All the Gram matrices are initialized to λI_d, where I_d denotes the d-dimensional identity matrix whereas the vectors $\mathbf{b}_{u,0}$ and $\theta_{u,0}$ are set to d-dimensional all-zeros vectors. At each round t, DILinUCB first uses the existing UCB estimates to compute the seed set $\mathcal{S}_t$ based on the given oracle ORACLE (line 4 of Algorithm 3). Then, it observes the pairwise reachability vector $\boldsymbol{y}_{u,t}$ for all the selected seeds in $\mathcal{S}_t$. The vector $\boldsymbol{y}_{u,t}$ is an n-dimensional column vector such that $\boldsymbol{y}_{u,t}(v) = \mathbb{1}\left(\{u\}, v, \mathcal{D}(\mathbf{w}_t)\right)$ indicating whether node v is reachable from the source u at round t. Finally, for each of the K selected seeds $u \in \mathcal{S}_t$, `DILinUCB` updates the sufficient statistics (lines 7 and 8 of Algorithm 3) and the UCB estimates (line 10 of Algorithm 3). Here, $\text{Proj}_{[0,1]}[\cdot]$ projects a real number onto the $[0, 1]$ interval.

2.4.5 Regret Bound

In this section, we derive a regret bound for `DILinUCB`, under (1) Assumption 1, (2) perfect linear generalization i.e. $q_{u,v} = \langle \theta^*_u, \mathbf{x}_v \rangle$ for all $u, v \in \mathcal{V}$, and (3) the assumption that $||\mathbf{x}_v||_2 \leq 1$ for all $v \in \mathcal{V}$. Notice that (2) is the standard assumption for linear bandit analysis (Dani et al., 2008), and (3) can always be satisfied by rescaling the target features. Our regret bound is stated below:

Theorem 3. *For any $\lambda, \sigma > 0$, any feature matrix X, any α-approximation oracle* `ORACLE`*, and any c satisfying*

$$c \geq \frac{1}{\sigma}\sqrt{dn \log\left(1 + \frac{nT}{\sigma^2 \lambda d}\right) + 2\log\left(n^2 T\right)} + \sqrt{\lambda} \max_{u \in \mathcal{V}} \|\theta^*_u\|_2, \tag{2.15}$$

if we apply `DILinUCB` *with input* $(\texttt{ORACLE}, X, c, \lambda, \sigma)$*, then its $\rho\alpha$-scaled cumulative regret is upper-bounded as*

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha} n^{\frac{3}{2}} \sqrt{\frac{dKT \log\left(1 + \frac{nT}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} + \frac{1}{\rho}. \tag{2.16}$$

For the tabular case $X = I$, we obtain a tighter bound

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha} n^{\frac{3}{2}} \sqrt{\frac{KT \log\left(1 + \frac{T}{\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} + \frac{1}{\rho}. \tag{2.17}$$

Recall that ρ specifies the quality of the surrogate approximation. Notice that if we choose $\lambda = \sigma = 1$, and choose c such that the inequality in 2.15 is tight, then our regret bound is $\widetilde{O}(n^2 d\sqrt{KT}/(\alpha\rho))$ for general feature matrix X, and $\widetilde{O}(n^{2.5}\sqrt{KT}/(\alpha\rho))$ in the tabular case. Here the $\widetilde{O}$ hides log factors. We now briefly discuss the tightness of our regret bounds. First, note that the $O(1/\rho)$ factor is due to the surrogate objective approximation discussed in Section 2.4.1, and the $O(1/\alpha)$ factor is due to the fact that `ORACLE` is an α-approximation algorithm. Second, note that the $\widetilde{O}(\sqrt{T})$-dependence on time is near-optimal, and the $\widetilde{O}(\sqrt{K})$-dependence on the cardinality of the seed sets is standard in the combinatorial semi-bandit literature (Kveton et al., 2015d). Third, for general X, notice that the $\widetilde{O}(d)$-dependence on feature dimension is standard in linear bandit literature (Dani et al., 2008; Wen et al., 2015b). To explain the $\widetilde{O}(n^2)$ factor in this case, notice that one $O(n)$ factor is due to the magnitude of the reward (the reward is from 0 to n, rather than 0 to 1), whereas one $\widetilde{O}(\sqrt{n})$ factor is due to the statistical dependence of the pairwise reachabilities. Assuming statistical independence between these reachabilities (similar to Chen et al. (2016b)), we can shave off this $\widetilde{O}(\sqrt{n})$ factor. However, this assumption is unrealistic in practice. Another $\widetilde{O}(\sqrt{n})$ is due to the fact that we learn one θ^*_u for each source node u (i.e. there is no generalization across the source nodes). Finally, for the tabular case $X = I$, the dependence on d no longer exists, but there is another $\widetilde{O}(\sqrt{n})$ factor due to the fact that there is no generalization across target nodes.

We conclude this section by sketching the proof for Theorem 3 (the detailed proof is available in Appendix A.5 and Appendix A.6). We define the "good event" as

$$\mathcal{F} = \{|\mathbf{x}_v^T(\widehat{\theta}_{u,t-1} - \theta^*_u)| \leq c\|\mathbf{x}_v\|_{\Sigma^{-1}_{u,t-1}} \, \forall u, v \in \mathcal{V}, \, t \leq T\},$$

and the "bad event" $\overline{\mathcal{F}}$ as the complement of $\mathcal{F}$. We then decompose the $\rho\alpha$-scaled regret $R^{\rho\alpha}(T)$ over $\mathcal{F}$ and $\overline{\mathcal{F}}$, and obtain the following inequality:

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\|\mathbf{x}_v\|_{\Sigma^{-1}_{u,t-1}}\middle|\mathcal{F}\right\} + \frac{P(\overline{\mathcal{F}})}{\rho}nT,$$

where $P(\overline{\mathcal{F}})$ is the probability of $\overline{\mathcal{F}}$. The regret bounds in Theorem 3 are derived based on worst-case bounds on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\|x_v\|_{\Sigma^{-1}_{u,t-1}}$ (Appendix A.5.2), and a bound on $P(\overline{\mathcal{F}})$ based on the "self-normalized bound for matrix-valued martingales" developed in Appendix A.6.

2.4.6 Practical Implementation

In this section, we briefly discuss how to implement our proposed algorithm, `DILinUCB`, in practical semi-bandit IM problems. Specifically, we will discuss how to construct features in Section 2.4.6, how to enhance the practical performance of `DILinUCB` based on Laplacian regularization in Section 2.4.6, and how to implement `DILinUCB` efficiently in real-world problems in Section 2.4.6.

Target Feature Construction

Although `DILinUCB` is applicable with any target feature matrix X, in practice, its performance is highly dependent on the "quality" of X. In this subsection, we motivate and propose a systematic feature construction approach based on the unweighted Laplacian matrix of the network topology $\mathcal{G}$. For all $u \in \mathcal{V}$, let $q_u \in \Re^n$ be the vector encoding the reachabilities from the seed u to all the target nodes $v \in \mathcal{V}$. Intuitively, q_u tends to be a smooth graph function in the sense that target nodes close to each other (e.g., in the same community) tend to have similar reachabilities from u. From (Belkin et al., 2006; Valko et al., 2014), we know that a smooth graph function (in this case, the reachability from a source) can be expressed as a linear combination of eigenvectors of the weighted Laplacian of the network. In our case, however, the edge weights correspond to influence probabilities and are unknown. However, we use the above intuition to construct target features based on the unweighted Laplacian of $\mathcal{G}$. Specifically, for a given $d = 1, 2, \ldots, n$, we set the feature matrix X to be the bottom d eigenvectors (associated with the smallest d eigenvalues) of the unweighted Laplacian of $\mathcal{G}$. Other approaches to construct target features include the neighbourhood preserving node-level features as described in (Grover and Leskovec, 2016; Perozzi et al., 2014). We leave the investigation of other feature construction approaches to future work.

Laplacian Regularization

One limitation of the proposed `DILinUCB` algorithm is that it does not share information across the seed nodes u. Specifically, it needs to learn the source node feature θ^*_u for each source node u independently, which might be inefficient for large-scale IM problems. Similar to target features, the source features also tend to be smooth in the sense that $\|\theta^*_{u_1} - \theta^*_{u_2}\|$ is "small" if nodes u_1 and u_2 are close to each other in the graph. We use this idea to design a prior which ties together the source features for different nodes, and hence transfers information between them. This idea of Laplacian regularization has been used in multi-task learning (Evgeniou et al., 2005) and for contextual-bandits in (Cesa-Bianchi et al., 2013; Vaswani et al., 2017b). Specifically, at each round t, we compute $\widehat{\theta}_{u,t}$

by minimizing the following objective:

$$\widehat{\theta}_{u,t} = \arg\min_{\theta_u} \left[\sum_{j=1}^{t} \sum_{u \in \mathcal{S}_t} (\mathbf{y}_{u,j} - X^T \theta_u)^2 + \lambda_2 \sum_{(u_1,u_2) \in \mathcal{E}} ||\theta_{u_1} - \theta_{u_2}||_2^2 \right]$$

where $\lambda_2 \geq 0$ is the regularization parameter. Further implementation details for this Laplacian regularization scheme are provided in Appendix A.3.

Computational Complexity

We now characterize the computational complexity of `DILinUCB`, and discuss how to implement it efficiently. Note that at each time t, `DILinUCB` needs to first compute a solution $\mathcal{S}_t$ based on `ORACLE`, and then update the UCBs. Since $\Sigma_{u,t}$ is positive semi-definite, the linear system in line 9 of Algorithm 3 can be solved using conjugate gradient in $O(\kappa d^2)$ time, where κ is the number of conjugate gradient iterations. It is straightforward to see the computational complexity to update the UCBs is $O(Knd^2)$. The computational complexity to compute $\mathcal{S}_t$ is dependent on `ORACLE`. For the classical setting in which $\mathcal{C} = \{\mathcal{S} \subseteq \mathcal{V} : |\mathcal{S}| \leq K\}$ and `ORACLE` is the greedy algorithm, the computational complexity is $O(Kn)$. To speed this up, we use the idea of lazy evaluations for submodular maximization proposed in (Minoux, 1978; Leskovec et al., 2007).

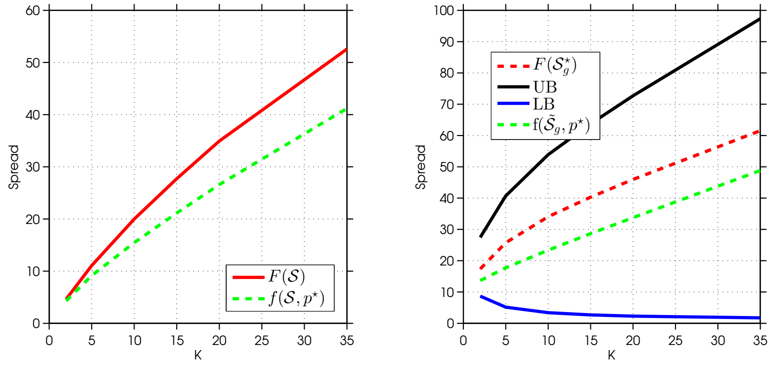
2.4.7 Experiments

In this section, we first empirically verify the quality of the surrogate objective and then evaluate the performance of `DILinUCB` on a real-world dataset.

Empirical Verification of Surrogate Objective

In this subsection, we empirically verify that the proposed surrogate $\widetilde{f}(\mathcal{S}, q)$ is a good approximation to the true IM objective $f(\mathcal{S})$. We conduct our tests on random Kronecker graphs, which are known to capture many properties of real-world social networks (Leskovec et al., 2010). Specifically, we generate a *social network instance* $(\mathcal{G}, \mathcal{D})$ as follows: we randomly sample $\mathcal{G}$ as a Kronecker graph with $n = 256$ and *sparsity* equal to 0.03 (Leskovec et al., 2005).[3] We choose $\mathcal{D}$ as the IC model and sample each of its influence probabilities

[3] Based on the sparsity of typical social networks.



(a) **(b)**

Figure 2.3: Experimental verification of surrogate objective.

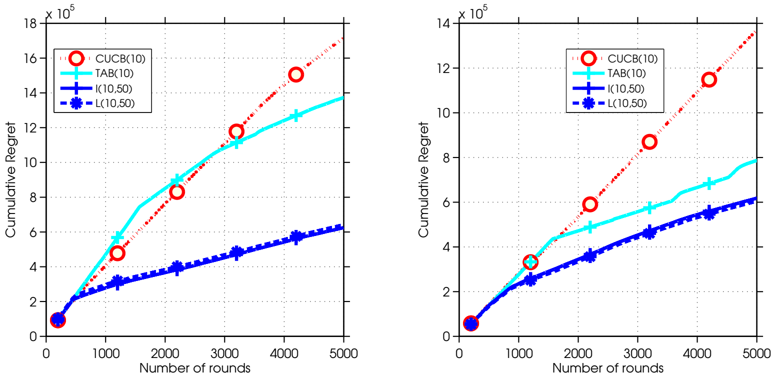
independently from the uniform distribution $U(0, 0.1)$. Note that this range of influence probabilities is guided by the empirical evidence in (Goyal et al., 2010; Barbieri et al., 2013). Note that all the results are averaged over 10 randomly generated instances.

We first numerically estimate the pairwise reachabilities q for each of the 10 instances based on social network simulation. In a simulation, we randomly sample a seed set $\mathcal{S}$ with cardinality K between 1 and 35, and record the pairwise influence indicator $\boldsymbol{y}_u(v)$ from each source $u \in \mathcal{S}$ to each target node v in this simulation. The reachability $q_{u,v}$ is estimated by averaging the $\boldsymbol{y}_u(v)$ values across 50k such simulations.

Based on the estimated values of q, we compare $\widetilde{f}(\mathcal{S}, q)$ and $f(\mathcal{S})$ as K, the seed set cardinality, varies from 2 to 35. For each K and each social network instance, we randomly sample 100 seed sets $\mathcal{S}$ with cardinality K. Then, we evaluate $\widetilde{f}(\mathcal{S}, q)$ based on the estimated q; and numerically evaluate $f(\mathcal{S})$ by averaging results of 500 influence simulations (diffusions). For each K, we average both $f(\mathcal{S})$ and $\widetilde{f}(\mathcal{S}, q)$ across the random seed sets in each instance as well as across the 10 instances. We plot the average $f(\mathcal{S})$ and $\widetilde{f}(\mathcal{S}, q)$ as a function of K in Figure 2.3a. The plot shows that $\widetilde{f}(\mathcal{S})$ is a good lower bound on the true expected spread $f(\mathcal{S})$, especially for low K.

Finally, we empirically quantify the surrogate approximation factor ρ. As before, we vary K from 2 to 35 and average across 10 instances. Let $\alpha = 1 - e^{-1}$. For each instance and each K, we first use the estimated q and the greedy algorithm to find an α-approximation solution $\widetilde{\mathcal{S}}_g$ to the surrogate problem in Equation 2.14. We then use the state-of-the-art IM algorithm (Tang et al., 2014) to compute an α-approximation solution $\mathcal{S}_g^*$ to the IM problem $\max_{\mathcal{S}} f(\mathcal{S})$. Since $f(\mathcal{S}_g^*) \geq \alpha f(\mathcal{S}^*)$ (Nemhauser et al., 1978), UB $\triangleq f(\mathcal{S}_g^*)/\alpha$ is an upper bound on $f(\mathcal{S}^*)$. From Theorem 2, LB $\triangleq f(\mathcal{S}_g^*)/K \leq f(\mathcal{S}^*)/K$ is a lower bound on $\widetilde{f}(\widetilde{\mathcal{S}}, q)$. We plot the average values (over 10 instances) of $f(\mathcal{S}_g^*)$, $\widetilde{f}(\widetilde{\mathcal{S}}_g, q)$, UB and LB against K in Figure 2.3b. We observe that the difference in spreads does not increase rapidly with K. Although ρ is lower-bounded with $\frac{1}{K}$, in practice for all $K \in [2, 35]$, $\rho \geq \frac{\alpha \widetilde{f}(\widetilde{\mathcal{S}}_g, q)}{f(\mathcal{S}_g^*)} \geq 0.55$. This shows that in practice, our surrogate approximation is reasonable even for large K. This can be explained as follows: because of the low influence probabilities in real-world networks, the probability that a node is influenced by a distant (in terms of graph distance) source node is extremely small and taking the max amongst the source nodes serves as a good approximation. This justifies the effectiveness of our surrogate approximation for real-world networks.

Performance of `DILinUCB`



(a) IC Model **(b)** LT Model

Figure 2.4: Comparing `DILinUCB` and `CUCB` on the Facebook subgraph with $K = 10$.

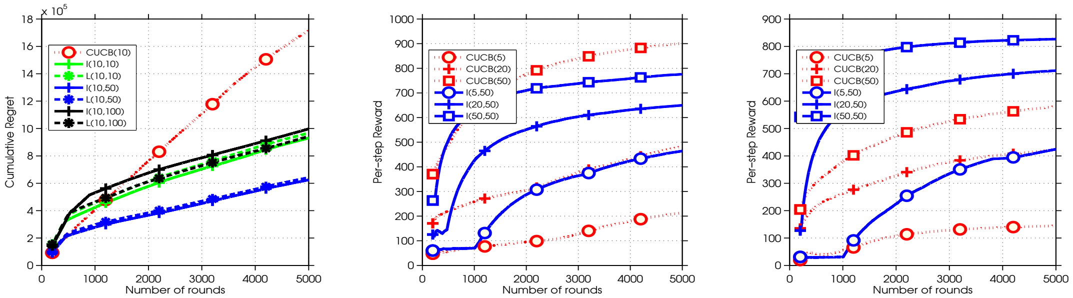
We now demonstrate the performance of variants of the `DILinUCB` algorithm and compare them with the state of the art. We choose the social network topology $\mathcal{G}$ as a subgraph of the Facebook network available at (Leskovec and Krevl, 2014), which consists of $n = 4$k nodes and $m = 88$k edges. Since the true diffusion model is unavailable, we assume the diffusion model $\mathcal{D}$ is either the independent cascade (IC) model or the linear threshold

(LT) model, and sample the edge influence probabilities independently from the uniform distribution $U(0, 0.1)$. We also choose $T = 5$k rounds.

We compare DILinUCB against the CUCB algorithm (Chen et al., 2016b) in both the IC model and the LT model, with $K = 10$. CUCB (referred to as CUCB(K) in plots) assumes the IC model, edge-level feedback and learns the influence probability for each edge independently. We demonstrate the performance of three variants of DILinUCB - the tabular case with $X = I$, independent estimation for each source node using target features (Algorithm 3) and Laplacian regularized estimation with target features (Section 2.4.6). In the subsequent plots, to emphasize the dependence on K and d, these are referred to as TAB(K), I(K,d) and L(K,d) respectively. We construct features as described in Section 2.4.6. Similar to spectral clustering (Von Luxburg, 2007), the gap in the eigenvalues of the unweighted Laplacian can be used to choose the number of eigenvectors d. In our case, we choose the bottom $d = 50$ eigenvectors for constructing target features and show the effect of varying d in the next experiment. Similar to (Gentile et al., 2014), all hyperparameters for our algorithm are set using an initial validation set of 500 rounds. The best validation performance was observed for $\lambda = 10^{-4}$ and $\sigma = 1$.

We now briefly discuss the performance metrics used in this section. For all $\mathcal{S} \subseteq \mathcal{V}$ and all $t = 1, 2 \ldots$, we define $r_t(\mathcal{S}) = \sum_{v \in \mathcal{V}} I\left(\mathcal{S}, v, \mathcal{D}(\mathbf{w}_t)\right)$, which is the realized reward at time t if $\mathcal{S}$ is chosen at that time. One performance metric is the *per-step reward*. Specifically, in one simulation, the per-step reward at time t is defined as $\frac{\sum_{s=1}^{t} r_s}{t}$. Another performance metric is the *cumulative regret*. Since it is computationally intractable to derive $\mathcal{S}^*$, our regret is measured with respect to $\mathcal{S}_g^*$, the α-approximation solution. In each simulation, the cumulative regret at round t is defined as $R(t) = \sum_{s=1}^{t} \left[r_s(\mathcal{S}_g^*) - r_s(\mathcal{S}_s) \right]$. All the subsequent results are averaged across 5 independent simulations.

Figures 2.4a and 2.4b show the cumulative regret when the underlying diffusion model is IC and LT, respectively. We have the following observations: (i) As compared to CUCB, the cumulative regret increases at a slower rate for all variants of DILinUCB, under both the IC and LT models, and for both the tabular case and the case with features. (ii) Exploiting target features (linear generalization) in DILinUCB leads to a much smaller cumulative regret. (iii) CUCB is not robust to model misspecification: it has a near linear cumulative regret under LT model. (iv) Laplacian regularization has little effect on the cumulative regret in these two cases. These observations clearly demonstrate the two main advantages of DILinUCB: it is both statistically efficient and robust to diffusion model misspecification.



(a) Effect of d in IC **(b)** Effect of K in IC **(c)** Effect of K in LT

Figure 2.5: Effects of varying d or K.

To explain (iv), we argue that the current combination of T, K, d and n results in sufficient feedback for independent estimation to perform well and hence it is difficult to observe any additional benefit of Laplacian regularization. We provide additional evidence for this argument in the next experiment.

In Figure 2.5a, we quantify the effect of varying d when the underlying diffusion model is IC and make the following observations: (i) The cumulative regret for both $d = 10$ and $d = 100$ is higher than that for $d = 50$. (ii) Laplacian regularization leads to observably lower cumulative regret when $d = 100$. Observation (iii) implies that $d = 10$ does not provide enough expressive power for linear generalization across the nodes of the network, whereas it is relatively difficult to estimate 100-dimensional θ_u^* vectors within 5k rounds. Observation (iv) implies that tying source node estimates together imposes an additional bias which becomes important while learning higher dimensional coefficients. This shows the potential benefit of using Laplacian regularization for larger networks, where we will need higher d for linear generalization across nodes. We obtain similar results under the LT model.

In Figures 2.5b and 2.5c, we show the effect of varying K on the per-step reward. We compare `CUCB` and the independent version of our algorithm when the underlying model is IC and LT. We make the following observations: (i) For both IC and LT, the per-step reward for all methods increases with K. (ii) For the IC model, the per-step reward for our algorithm is higher than `CUCB` when $K = \{5, 10, 20\}$, but the difference in the two spreads decreases with K. For $K = 50$, `CUCB` outperforms our algorithm. (iii) For the LT model, the per-step reward of our algorithm is substantially higher than `CUCB` for all

K. Observation (i) is readily explained since both IC and LT are progressive models, and satisfy Assumption 1. To explain (ii), note that `CUCB` is correctly specified for the IC model. As K becomes higher, more edges become active and `CUCB` observes more feedback. It is thus able to learn more efficiently, leading to a higher per-step reward compared to our algorithm when $K = 50$. Observation (iii) again demonstrates that `CUCB` is not robust to diffusion model misspecification, while `DILinUCB` is.

2.5 Related Work

The IMB problem has been studied in several recent papers (Wang and Chen, 2017; Chen et al., 2016b; Vaswani et al., 2015; Carpentier and Valko, 2016). In (Chen et al., 2016b), Chen *et al* studied it under edge semi-bandit feedback and the IC diffusion model. They formulated it as a combinatorial multi-armed bandit problem and proposed the `CUCB` algorithm. However, their bounds on the cumulative regret depend on the reciprocal of the minimum observation probability of the edges and can be exponentially high. For example, consider a line graph with m edges where all edge weights are 0.5. Then the minimum observation probability is 2^{m-1}. In contrast, our derived regret bounds are polynomial in all quantities of interest.

A recent result of Wang and Chen (Wang and Chen, 2017) removes this dependence on the minimum observation probability in the tabular case. They present a worst-case bound of $\widetilde{\mathcal{O}}(nm\sqrt{T})$, which in the case of a complete graph, improves our result by $\widetilde{\mathcal{O}}(n)$. Unlike us, their analysis does not depend on the structural properties of the network. Moreover, both Chen et al. (2016a) and Wang and Chen (2017) do not consider generalization across edges or nodes, and therefore their proposed algorithms are unlikely to be practical for real-world social networks. Vaswani *et al* (Vaswani et al., 2015) address the IMB problem under the more challenging node semi-bandit feedback model but they do not give any theoretical guarantees. Moreover, all of the above work assumes the independent cascade model of diffusion. In contrast, we also consider a model-agnostic setting for the IMB problem.

Carpentier and Valko (Carpentier and Valko, 2016) give a minimax optimal algorithm for IM bandits under a *local* model of influence where only the immediate neighbours of a seed node can be influenced. In the related work on networked bandits (Fang and Tao, 2014), the learner chooses a node and its reward is the *sum* of the rewards of the chosen node and its immediate neighbourhood. In (Lei et al., 2015), Lei *et al*consider the related, but different problem of maximizing the number of unique activated nodes across multiple rounds. The algorithms do not have theoretical guarantees and do not consider any generalization model across nodes or edges. Lagrée *et al* (Lagrée et al., 2017) consider a *persistent* extension of IM bandits when some nodes become persistent over the rounds and no longer yield rewards. Singla et al. (2015) considers the IM setting with additional observability constraints, where we face a restriction on which nodes we can choose at each round. This setting is related to the *volatile multi-armed bandits* where the set of possible arms changes (Bnaya et al., 2013) across rounds.

Furthermore the IMB problem is also a generalization and extension of recent work on cascading bandits (Kveton et al., 2015a,b), since cascading bandits can be viewed as variants of online influence maximization problems with special topologies (chains).

2.6 Conclusion and Future Work

In the first part of this chapter, we studied the IMB problem under the independent cascade model and edge semi-bandit feedback. In the second part, we developed a novel parametrization for IMB that enables our framework to be agnostic to the underlying model of diffusion. This parametrization allows us to use a weaker model of feedback from the network, while retaining the ability to learn in a statistically efficient manner. For each of these settings, we proposed a UCB-based algorithm, analysed it theoretically and empirically verified its effectiveness.

Our IMB framework can be easily extended to the contextual bandit setting where the activation probabilities depend on the context of the product being marketed. In the future, it would be interesting to experiment with alternate bandit algorithms such Thompson sampling, and feedback models such as the node semi-bandit feedback in Vaswani et al. (2015). We also plan to conduct an extensive empirical study to test the effectiveness of the algorithms proposed in this chapter on large real-world datasets.

Chapter 3

Content-based Recommendation

In this chapter, we use the contextual bandit framework for content-based recommendation in the presence of a user-user network.

3.1 Introduction

Let us consider a newly established recommender system that has little or no information about the preferences of its users. Since it has not collected enough rating data from the users, it is unable to use traditional collaborative filtering based methods (Su and Khoshgoftaar, 2009) to infer the users' preferences in order to make good recommendations. Such a scenario, known as the *cold-start* problem in the recommender system (RS) literature, is especially common for newly formed E-commerce or social media companies. One approach for addressing this problem is to adopt the bandit framework (Li et al., 2010), wherein the new system attempts to learn the users' preferences while simultaneously making recommendations to them.

Let us first map this problem to the generic bandit framework described in Algorithm 1. In this case, the RS is the agent making decisions about recommendations, the environment consists of the system's users and a possible action is recommending an item to a particular target user. The feedback consists of the rating given by the user to the recommended product. The agent chooses an item to recommend (SELECT), receives a corresponding rating from the user (OBSERVE) and revises its estimation of the user's preferences (UPDATE). In this case, exploration consists of recommending items that have not been rated or seen by a particular user in order to better learn their preferences. At the same time, the RS should recommend "relevant" items that will be liked by and elicit higher ratings from its users and this constitutes exploitation.

Since the number of available items (arms in this case) is large, it is useful to share information to quickly infer a user's preferences for similar items. To model this, we assume that each item can be described by its content; for example, a set of tags or keywords describing a news article or a movie. An additional complication in the scenario of personalized news recommendation or in recommending trending Facebook posts is that the set of available items is not fixed but changing continuously. To handle these challenges, previous work in (Li et al., 2010) (Li et al., 2011) makes use of the contextual bandit framework described in Chapter 1.

However, this framework assumes that users interact with the RS in an isolated manner, when in fact the RS might have an associated social component. This has become increasingly common and popular sites such as Goodreads, Quora and Facebook are a few examples where a recommender system has an associated social network of users. Instead of learning the preferences of the large number of users in isolation, the basic idea is to leverage the relationships between them in order to facilitate learning with fewer interactions.

A recent approach that leverages a social network of users to improve recommendations is the gang of bandits (GOB) model (Cesa-Bianchi et al., 2013). In particular, the GOB model exploits the homophily effect (McPherson et al., 2001) that suggests users with similar preferences are more likely to form links in a social network. It models the social network as a graph where the nodes correspond to users and the edges correspond to relationships (friendship on Facebook or "following" on Twitter). Given this graph, homophily implies that user preferences vary smoothly across the social graph and tend to be similar for users connected with each other. This assumption allows us to transfer information between users implying that we can learn about a user from his or her friends' ratings. However, the recommendation algorithm proposed in (Cesa-Bianchi et al., 2013) has a computational complexity *quadratic* in the number of nodes and thus can only be used for networks with a small number of users. Several recent works have tried to improve the scaling of the GOB model by clustering the users into groups (Gentile et al., 2014; Nguyen and Lauw, 2014), but this approach limits the flexibility of the model and loses the ability to model individual users' preferences.

In Section 3.2, we cast the problem in the framework of Gaussian Markov random fields (GMRFs). This connection enables us to scale the GOB framework to much larger graphs, while retaining the ability to model individual users. Specifically, we interpret the GOB model as the optimization of a Gaussian likelihood on the users' observed ratings and interpret the user-user graph as the prior inverse-covariance matrix of a GMRF. From this perspective, we can efficiently estimate the users' preferences by performing MAP estimation in a GMRF. In Section 3.3, we propose a Thompson sampling algorithm that exploits the recent sampling-by-perturbation idea from the GMRF literature (Papandreou and Yuille, 2010) to scale to even larger problems. This idea is fairly general and might be of independent interest in the efficient implementation of Thompson sampling methods. We establish regret bounds for Thompson sampling as well as an ε-greedy strategy. Our theoretical bounds show that using the user-user graph can provably lead to a lower cumulative regret. Experimental results in Section 3.4 indicate that our methods are as good as or significantly better than approaches which ignore the graph or those that cluster the nodes. Finally, when the graph of users is not available, we propose a heuristic for learning the graph and user preferences simultaneously in an alternating minimization framework detailed in Appendix B. We conclude this chapter by surveying the related work in Section 3.5 and giving some ideas for future research in Section 3.6.

3.2 Scaling up Gang of Bandits

In this section, we first describe the general GOB framework, then discuss the relationship to GMRFs, and finally show how this leads to more scalable method. In this chapter, $\text{Tr}(A)$ denotes the trace of matrix A, $A \otimes B$ denotes the Kronecker product of matrices A and B, I_d refers to the d-dimensional identity matrix, and $\text{vec}(A)$ is the operation of stacking the columns of a matrix A into a vector.

3.2.1 Gang of Bandits Framework

Recall that in the contextual bandit framework, a set of features $\mathcal{C}_t = [\mathbf{x}_{1,t}, \mathbf{x}_{2,t} \dots \mathbf{x}_{K,t}]$ becomes available in each round t. In the recommendation setting, the set $\mathcal{C}_t$ refers to the features for the available items at round t. These might be features corresponding to movies released in a particular week, news articles published on a particular day, or trending stories on Facebook. For ease of notation, when the round is fixed and implied, we use $\mathbf{x}_j$ to refer to the feature vector for item j in round t. We denote the number of users by n

and assume that $|\mathcal{C}_t| = K$ for all t. Furthermore, we denote the (unknown) ground-truth vector of preferences for user i as $\theta_i^* \in \mathbb{R}^d$. The user to whom a recommendation is being made in round t is referred to as the *target user* and is denoted by i_t. [1]

Given the target user i_t, the RS recommends an available item $j_t \in \mathcal{C}_t$ to them. The user i_t then provides feedback on the recommended item j_t in the form of a rating r_{i_t,j_t}. Based on this feedback, the estimated preference vector for user i_t is updated. In this case, the cumulative regret measures the loss in recommendation performance due to lack of knowledge of the users' preferences. In particular, the expected cumulative regret $\mathbb{E}[R(T)]$ after T rounds is given by:

$$\mathbb{E}[R(T)] = \sum_{t=1}^{T} \left[\max_{\mathbf{x}_j \in \mathcal{C}_t} \langle \theta_{i_t}^*, \mathbf{x}_j \rangle - \langle \theta_{i_t}^*, \mathbf{x}_{j_t,t} \rangle \right]. \tag{3.1}$$

We make the following assumptions for our analysis:

Assumption 3. *The ℓ_2-norms of the true preference vectors and item feature vectors are bounded from above. Without loss of generality we assume that $||x_j||_2 \leq 1$ for all j and $||\theta_i^*||_2 \leq 1$ for all i. Also without loss of generality, we assume that the ratings are in the range $[0, 1]$.*

Assumption 4. *The true ratings can be given by a linear model (Li et al., 2010), meaning that $r_{i,j} = \langle \theta_i^*, \mathbf{x}_j \rangle + \eta_{i,j}$ for some noise term $\eta_{i,j}$.*

These are standard assumptions in the literature. We denote the history of observations until round t as $\mathbb{H}_{t-1} = \{(i_\tau, j_\tau, r_{i_\tau,j_\tau})\}_{\tau=1,2\cdots t-1}$ and the union of the set of available items until round t along with their corresponding features as $\mathbb{C}_{t-1}$.

Assumption 5. *The noise $\eta_{i,j}$ is conditionally sub-Gaussian (Agrawal and Goyal, 2012b)(Cesa-Bianchi et al., 2013) with zero mean and bounded variance, meaning that $\mathbb{E}[\eta_{i,j} \mid \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] = 0$ and that there exists a $\sigma > 0$ such that for all $\gamma \in \mathbb{R}$, we have $\mathbb{E}[\exp(\gamma \eta_{i,j}) \mid \mathbb{H}_{t-1}, \mathbb{C}_{t-1}] \leq \exp(\frac{\gamma^2 \sigma^2}{2})$.*

This assumption implies that for all i and j, the conditional mean is given by $\mathbb{E}[r_{i,j} | \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] = \langle \theta_i^*, \mathbf{x}_j \rangle$ and that the conditional variance satisfies $\mathbb{V}[r_{i,j} | \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] \leq \sigma^2$.

In the GOB framework, we assume access to a (fixed) graph $G = (\mathcal{V}, \mathcal{E})$ of users in the form of a social network (or a "trust graph"). Here, the nodes $\mathcal{V}$ correspond to users,

[1] Throughout the paper, we assume there is only a single target user per round. It is straightforward extend our results to multiple target users.

whereas the edges $\mathcal{E}$ correspond to friendships or trust relationships. The homophily effect implies that the true user preferences vary smoothly across the graph, so we expect the preferences of users connected in the graph to be close to each other. Specifically, we make the following assumption:

Assumption 6. *The true user preferences vary smoothly according to the given graph, in the sense that we have a small value of the quantity*

$$\sum_{(i_1,i_2)\in\mathcal{E}} ||\theta^*_{i_1} - \theta^*_{i_2}||^2.$$

In other words, we assume that the graph acts as a correctly-specified prior on the users' true preferences. Note that this assumption implies that nodes in dense subgraphs will have a higher similarity than those in sparse subgraphs (since they will have a larger number of neighbours).

This assumption can be violated in some datasets. For example, in our experiments we consider one dataset in which the available graph is imperfect, in that user preferences do not seem to vary smoothly across all graph edges. Intuitively, we might think that transferring information between users might be harmful in this case (compared to ignoring the graph structure). However, in our experiments, we observe that even in these cases, the GOB approach still lead to results as good as ignoring the graph.

The GOB model in (Cesa-Bianchi et al., 2013) solves a contextual bandit problem for each user, where the mean vectors in the different problems are related according to the Laplacian of the graph. If A is the adjacency matrix for the graph G and D is the diagonal matrix of node degrees in the graph, then the normalized Laplacian $L = I_d - D^{-1/2}AD^{-1/2}$. In practice, to ensure invertibility, we add the identity matrix I_n to the normalized Laplacian.

Let $\theta_{i,t}$ be the preference vector estimate for user i at round t. Let θ_t and $\theta^* \in \mathbb{R}^{dn}$ (respectively) be the concatenation of the vectors $\theta_{i,t}$ and θ^*_i across all users. Note that Assumption 6 implies that the term $\theta^\mathsf{T}(L \otimes I_d)\theta$ should be small. The GOB model thus solves the following regression problem to estimate the mean preference vector at round t,

$$\theta_t = \arg\min_{\theta} \left[\sum_{i=1}^{n} \sum_{k\in\mathcal{M}_{i,t}} (\langle \theta_i, \mathbf{x}_k \rangle - r_{i,k})^2 + \lambda\theta^\mathsf{T}(L \otimes I_d)\theta \right], \tag{3.2}$$

where $\mathcal{M}_{i,t}$ is the set of items rated by user i up to round t.

The first term is a data-fitting term and models the observed ratings. The second term is the Laplacian regularization equal to $\sum_{(i,j)\in\mathcal{E}} \lambda ||\theta_{i,t} - \theta_{j,t}||_2^2$. This term models smoothness across the graph and $\lambda > 0$ is a tunable hyper-parameter that controls the strength of this regularization. Note that the same objective function has also been explored for graph-regularized multi-task learning in (Evgeniou and Pontil, 2004).

3.2.2 Connection to GMRFs

Unfortunately, for the approach proposed in (Cesa-Bianchi et al., 2013), solving Equation 3.2 has a computational complexity of $O(d^2n^2)$. To solve it more efficiently, we now show that the above optimization problem can be interpreted as performing MAP estimation in a GMRF. This will allow us to apply the GOB model to much larger datasets, and lead to an even more scalable algorithm based on Thompson sampling (Section 3.3).

Consider the following generative model for the ratings $r_{i,j}$ and the user preference vectors θ_i,

$$r_{i,j} \sim \mathcal{N}(\langle \theta_i, \mathbf{x}_j \rangle, \sigma^2), \quad \theta \sim \mathcal{N}(0, (\lambda L \otimes I_d)^{-1}).$$

This GMRF model assumes that the ratings $r_{i,j}$ are independent given θ_i and $\mathbf{x}_j$, which is the standard regression assumption. Under this assumption, the first term in Equation 3.2 is equal to the negative log-likelihood for all of the observed ratings $\mathbf{r}_t$ at time t, $\log \mathcal{P}(\mathbf{r}_t \mid \theta, \mathbf{x}_t, \sigma)$, up to an additive constant and assuming $\sigma = 1$. Similarly, the negative log-prior $\mathcal{P}(\theta \mid \lambda, L)$ in this model gives the second term in Equation 3.2 (again, up to an additive constant that does not depend on θ). Thus, by Bayes rule minimizing the objective in Equation 3.2 is equivalent to maximizing the posterior in this GMRF model.

To characterize the posterior, it is helpful to introduce the notation $\boldsymbol{\phi}_{i,j} \in \mathbb{R}^{dn}$ to represent the "global" feature vector corresponding to recommending item j to user i. In particular, let $\boldsymbol{\phi}_{i,j}$ be the concatenation of n d-dimensional vectors where the i^{th} vector is equal to $\mathbf{x}_j$ and the others are zero. The rows of the $t \times dn$ dimensional matrix Φ_t correspond to these "global" features for all the recommendations made until time t. Under this notation, the posterior $p(\theta \mid \mathbf{r}_t, \theta, \Phi_t)$ is given by a $\mathcal{N}(\theta_t, \Sigma_t^{-1})$ distribution with $\Sigma_t = \frac{1}{\sigma^2}\Phi_t^\intercal \Phi_t + \lambda(L \otimes I_d)$ and $\theta_t = \frac{1}{\sigma^2}\Sigma_t^{-1}\mathbf{b}_t$ with $\mathbf{b}_t = \Phi_t^\intercal \mathbf{r}_t$. We can view the approach in (Cesa-Bianchi et al., 2013) as explicitly constructing the dense $dn \times dn$ matrix Σ_t^{-1}, leading to an $O(d^2n^2)$ memory requirement. A new recommendation at round t is thus equivalent to a rank-1 update to Σ_t, and even with the Sherman-Morrison formula this leads to an $O(d^2n^2)$ time requirement in each iteration.

3.2.3 Scalability

Rather than treating Σ_t as a general matrix, we propose to exploit its structure to scale up the GOB framework to problems where n is very large. In particular, solving Equation 3.2 corresponds to finding the mean vector of the GMRF, which corresponds to solving the linear system $\Sigma_t\theta = \mathbf{b}_t$. Since Σ_t is positive-definite, the linear system can be solved using conjugate gradient (Hestenes and Stiefel, 1952). Conjugate gradient notably does not require Σ_t^{-1}, but instead uses matrix-vector products $\Sigma_t\mathbf{v} = (\Phi_t^\intercal\Phi_t)\mathbf{v} + \lambda(L \otimes I_d)\mathbf{v}$ for vectors $\mathbf{v} \in \mathbb{R}^{dn}$. Note that $\Phi_t^\intercal\Phi_t$ is block diagonal and has only $O(nd^2)$ non-zeroes. Hence, $\Phi_t^\intercal\Phi_t\mathbf{v}$ can be computed in $O(nd^2)$ time. For computing $(L \otimes I_d)\mathbf{v}$, we use the fact that $(B^\intercal \otimes A)\mathbf{v} = \text{vec}(AVB)$, where V is an $n \times d$ matrix such that $\text{vec}(V) = \mathbf{v}$. This implies $(L \otimes I_d)\mathbf{v}$ can be written as $VL^\intercal$ which can be computed in $O(d \cdot \text{nnz}(L))$ time, where $\text{nnz}(L)$ is the number of non-zeroes in L and is equal to the number of edges in the graph. This approach thus has a memory requirement of $O(nd^2 + \text{nnz}(L))$ and a time complexity of $O(\kappa(nd^2 + d \cdot \text{nnz}(L)))$ per mean estimation. Here, κ is the number of conjugate gradient iterations which depends on the condition number of the matrix (we used warm-starting by the solution in the previous round for our experiments, which meant that $\kappa = 5$ was enough for convergence). Thus, the algorithm scales linearly in n and in the number of edges of the network (which tends to be linear in n due to the sparsity of social relationships). This enables us to scale to large networks, of the order of 50K nodes and millions of edges.

3.3 Alternative Bandit Algorithms

The above structure can be used to speed up the mean estimation for any algorithm in the GOB framework. However, the LINUCB-like algorithm in (Cesa-Bianchi et al., 2013) needs to estimate the confidence intervals $\sqrt{\boldsymbol{\phi}_{i,j}^\intercal \Sigma_t^{-1} \boldsymbol{\phi}_{i,j}}$ for each available item j in the set $\mathcal{C}_t$. Using the above scalability idea, estimating these requires $O(|\mathcal{C}_t|\kappa(nd^2 + d \cdot \text{nnz}(L)))$ time since we need solve the linear system with $|\mathcal{C}_t|$ right-hand sides, one for each available item. But this becomes impractical when the number of available items in each round is large.

We propose two approaches for mitigating this: first, in this section we adapt the Epoch-greedy (Langford and Zhang, 2008) algorithm to the GOB framework. We also propose a GOB variant of Thompson sampling (Li et al., 2010) and further exploit the connection to GMRFs to scale it to even larger problems by using the recent sampling-by-perturbation trick (Papandreou and Yuille, 2010). This GMRF connection and scalability trick might be of independent interest for Thompson sampling in other large-scale problems.

3.3.1 Epoch-Greedy

Epoch-greedy (Langford and Zhang, 2008) is a variant of the popular ε-greedy algorithm that explicitly differentiates between exploration and exploitation rounds. In this case, an "exploration" round consists of recommending a random item from $\mathcal{C}_t$ to the target user i_t. The feedback from these exploration rounds is used to learn θ^*. An "exploitation" round consists of choosing the available item $\widehat{j_t}$ which maximizes the expected rating, $j_t = \arg\max_{j \in \mathcal{C}_t} \widehat{\theta_t}^{\mathsf{T}} \boldsymbol{\phi}_{i_t,j}$. Epoch-greedy proceeds in epochs, where each epoch q consists of 1 exploration round and s_q exploitation rounds.

Scalability: The time complexity for Epoch-Greedy is dominated by the exploitation rounds that require computing the mean and estimating the expected rating for all the available items. Given the mean vector, this estimation takes $O(d|\mathcal{C}_t|)$ time. The overall time complexity per exploitation round is thus $O(\kappa(nd^2 + d \cdot \text{nnz}(L)) + d|\mathcal{C}_t|)$.

Regret: We assume that we incur a maximum regret of 1 in an exploration round, whereas the regret incurred in an exploitation round depends on how well we have learned θ^*. The attainable regret is thus proportional to the generalization error for the class of hypothesis functions mapping the context vector to an expected rating (Langford and Zhang, 2008). In our case, the class of hypotheses is a set of linear functions (one for each user) with Laplacian regularization. We characterize the generalization error in the GOB framework in terms of its Rademacher complexity (Maurer, 2006), and use this to bound the expected regret. For ease of exposition in the regret bounds, we suppress the factors that don't depend on either n, L, λ or T. The complete bound is stated in Appendix B.

Theorem 4. *Under the additional assumption that (a) $||\theta_t||_2 \leq 1$ for all rounds t and that (b) the regularization parameter λ is a constant that can not depend on the horizon T, the expected regret obtained by epoch-greedy in the GOB framework is given as:*

$$R(T) = \widetilde{O}\left(n^{1/3}\left(\frac{\text{Tr}(L^{-1})}{\lambda n}\right)^{\frac{1}{3}} T^{\frac{2}{3}}\right)$$

Proof Sketch. Let $\mathcal{H}$ be the class of valid hypotheses of linear functions (the preference vector for each user i) coupled with Laplacian regularization (because of the network structure). Let $Err(q, \mathcal{H})$ be the generalization error for $\mathcal{H}$ after obtaining q unbiased samples in the exploration rounds. We adapt Corollary 3.1 from (Langford and Zhang, 2008) to our context:

Lemma 1. *If* $s_q = \left\lfloor \frac{1}{Err(q,\mathcal{H})} \right\rfloor$ *and* Q_T *is the smallest* Q *such that* $Q + \sum_{q=1}^{Q} s_q \geq T$*, the regret obtained by Epoch-Greedy can be bounded as* $R(T) \leq 2Q_T$*.*

We use the result in (Maurer, 2006) to bound the generalization error of our class of hypotheses in terms of its empirical Rademacher complexity $\widehat{\mathcal{R}}_q^n(\mathcal{H})$. With probability $1-\delta$,

$$Err(q, \mathcal{H}) \leq \widehat{\mathcal{R}}_q^n(\mathcal{H}) + \sqrt{\frac{9\ln(2/\delta)}{2q}}. \tag{3.3}$$

Using Theorem 2 in (Maurer, 2006) and Theorem 12 from (Bartlett and Mendelson, 2003), we obtain

$$\widehat{\mathcal{R}}_q^n(\mathcal{H}) \leq \frac{2}{\sqrt{q}}\sqrt{\frac{12Tr(L^{-1})}{\lambda}}. \tag{3.4}$$

Using (3.3) and (3.4) we obtain

$$Err(q, \mathcal{H}) \leq \frac{\left[2\sqrt{12Tr(L^{-1})/\lambda} + \sqrt{\frac{9\ln(2/\delta)}{2}}\right]}{\sqrt{q}}. \tag{3.5}$$

The theorem follows from (3.5) along with Lemma 1. □

Note that this theorem assumes that preference vectors θ are smooth according to the given graph structure. The effect of the graph in the regret bound is reflected through the term $\mathrm{Tr}(L^{-1})$. For a connected graph, we have the following upper-bound $\frac{\mathrm{Tr}(L^{-1})}{n} \leq \frac{(1-1/n)}{\nu_2} + \frac{1}{n}$ (Maurer, 2006). Here, ν_2 is the second smallest eigenvalue of the Laplacian. The value ν_2 represents the algebraic connectivity of the graph (Fiedler, 1973). For a more connected graph, ν_2 is higher, the value of $\frac{\mathrm{Tr}(L^{-1})}{n}$ is lower, resulting in a smaller regret. Note that although this result leads to a sub-optimal dependence on T ($T^{\frac{2}{3}}$ instead of $T^{\frac{1}{2}}$), our experiments incorporate a small modification that gives similar performance to the more-expensive algorithm in (Cesa-Bianchi et al., 2013).

3.3.2 Thompson sampling

A common alternative to LINUCB and Epoch-Greedy is Thompson sampling (TS). In this case, each iteration TS uses a sample $\widetilde{\theta}_t$ from the posterior distribution at round t, $\widetilde{\theta}_t \sim \mathcal{N}(\theta_t, \Sigma_t^{-1})$. It then selects the item based on the obtained sample, $j_t = \arg\max_{j \in \mathcal{C}_t} \widetilde{\theta}_t^{\mathsf{T}} \boldsymbol{\phi}_{i_t,j}$. We show below that the GMRF connection makes TS scalable, but unlike Epoch-Greedy it also achieves the optimal regret.

Scalability: The conventional approach for sampling from a multivariate Gaussian posterior involves forming the Cholesky factorization of the posterior covariance matrix. But in the GOB model the posterior covariance matrix is a dn-dimensional matrix where the fill-in from the Cholesky factorization can lead to a computational complexity of $O(d^2n^2)$. In order to implement Thompson sampling for large networks, we adapt the recent sampling-by-perturbation approach (Papandreou and Yuille, 2010) to our setting, and this allows us to sample from a Gaussian prior and then solve a linear system to sample from the posterior.

Let $\widetilde{\theta}_0$ be a sample from the prior distribution and let $\widetilde{\mathbf{r}}_t$ be the perturbed (with standard normal noise) rating vector at round t, meaning that $\widetilde{\mathbf{r}}_t = \mathbf{r}_t + \mathbf{y}_t$ for $\mathbf{y}_t \sim \mathcal{N}(0, I_t)$. In order to obtain a sample $\widetilde{\theta}_t$ from the posterior, we can solve the linear system

$$\Sigma_t \widetilde{\theta}_t = (L \otimes I_d)\widetilde{\theta}_0 + \Phi_t^T \widetilde{\mathbf{r}}_t. \tag{3.6}$$

Let S be the Cholesky factor of L so that $L = SS^T$. Note that $L \otimes I_d = (S \otimes I_d)(S \otimes I_d)^T$. If $\mathbf{z} \sim \mathcal{N}(0, I_{dn})$, we can obtain a sample from the prior by solving $(S \otimes I_d)\widetilde{\theta}_0 = \mathbf{z}$. Since S tends to be sparse (using for example (Davis, 2005; Kyng and Sachdeva, 2016)), this equation can be solved efficiently using conjugate gradient. We can pre-compute and store S and thus obtain a sample from the prior in time $O(d \cdot \text{nnz}(L))$. Using that $\Phi_t^T \widetilde{\mathbf{r}}_t = \mathbf{b}_t + \Phi_t^T \mathbf{y}_t$ in (3.6) and simplifying we obtain

$$\Sigma_t \widetilde{\theta}_t = (L \otimes I_d)\widetilde{\theta}_0 + \mathbf{b}_t + \Phi_t^T \mathbf{y}_t \tag{3.7}$$

As before, this system can be solved efficiently using conjugate gradient. Note that solving (3.7) results in an exact sample from the dn-dimensional posterior. Computing $\Phi_t^T \mathbf{y}_t$ has a time complexity of $O(dt)$. Thus, this approach is faster than the original GOB framework whenever $t < dn^2$. Since we focus on the case of large graphs, this condition will tend to hold in our setting.

We now describe an alternative method for constructing the right side of Equation 3.7 that doesn't depend on t. Observe that computing $\Phi_t^T \mathbf{y}_t$ is equivalent to sampling from the distribution $\mathcal{N}(0, \Phi_t^T \Phi_t)$. To sample from this distribution, we maintain the Cholesky factor P_t of $\Phi_t^T \Phi_t$. Recall that the matrix $\Phi_t^T \Phi_t$ is block diagonal (one block for every user) for all rounds t. Hence, its Cholesky factor P_t also has a block diagonal structure and requires $\mathcal{O}(nd^2)$ storage. In each round, we make a recommendation to a single user and thus make a rank-1 update to only one $d \times d$ block of P_t. This is an order $\mathcal{O}(d^2)$ operation. Once we have an updated P_t, sampling from $\mathcal{N}(0, \Phi_t^T \Phi_t)$ and constructing the right side of (3.7) is an $O(nd^2)$ operation. The per-round computational complexity for our TS approach is thus $O(\min\{nd^2, dt\} + d \cdot nnz(L))$ for forming the right side in (3.7), $O(nd^2 + d \cdot \mathrm{nnz}(L))$ for solving the linear system in (3.7) as well as for computing the mean, and $O(d \cdot |\mathcal{C}_t|)$ for selecting the item. Thus, our proposed approach has a complexity linear in the number of nodes and edges and can scale to large networks.

Regret: To analyze the regret with TS, observe that TS in the GOB framework is equivalent to solving a single dn-dimensional contextual bandit problem, but with a modified prior covariance equal to $(\lambda L \otimes I_d)^{-1}$ instead of I_{dn}. We obtain the result below by following a similar argument to Theorem 1 in (Agrawal and Goyal, 2012b). The main challenge in the proof is to make use of the available graph to bound the variance of the arms. We first state the result and then sketch the main differences from the original proof.

Theorem 5. *Under the following additional technical assumptions: (a)* $\log(K) < (dn - 1)\ln(2)$*, (b)* $\lambda < dn$*, and (c)* $\log\left(\frac{3+T/\lambda dn}{\delta}\right) \le \log(KT)\log(T/\delta)$*, with probability* $1-\delta$*, the regret obtained by Thompson Sampling in the GOB framework is given as:*

$$R(T) = \widetilde{O}\left(\frac{dn\sqrt{T}}{\sqrt{\lambda}} \sqrt{\log\left(\frac{3\,\mathrm{Tr}(L^{-1})}{n} + \frac{\mathrm{Tr}(L^{-1})T}{\lambda dn^2\sigma^2}\right)}\right)$$

Proof Sketch. To make the notation cleaner, for the round t and target user i_t under consideration, we use j to index the available items. Let the index of the optimal item at round t be j^*_t whereas the index of the item chosen by our algorithm is denoted j_t. Let $s_t(j)$ be the standard deviation in the estimated rating of item j at round t. It is given as $s_t(j) = \sqrt{\boldsymbol{\phi}_j^{\mathsf{T}} \Sigma_{t-1}^{-1} \boldsymbol{\phi}_j}$. Further, let $l_t = \sqrt{dn \log\left(\frac{3+t/\lambda dn}{\delta}\right)} + \sqrt{3\lambda}$. Let $\mathcal{E}^{\mu}(t)$ be the event such that for all j,

$$\mathcal{E}^{\mu}(t): \quad |\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| \leq l_t s_t(j)$$

We first prove that, for $\delta \in (0,1)$, $p(\mathcal{E}^{\mu}(t)) \geq 1 - \delta$.

Define $g_t = \sqrt{4 \log(tK)}\rho_t + l_t$, where $\rho_t = \sqrt{9d \log\left(\frac{t}{\delta}\right)}$. Let $\gamma = \frac{1}{4e\sqrt{\pi}}$. Given that the event $\mathcal{E}^{\mu}(t)$ holds with high probability, we follow an argument similar to Lemma 4 of (Agrawal and Goyal, 2012b) and obtain the following bound:

$$R(T) \leq \frac{3g_T}{\gamma} \sum_{t=1}^{T} s_t(j_t) + \frac{2g_T}{\gamma} \sum_{t=1}^{T} \frac{1}{t^2} + \frac{6g_T}{\gamma} \sqrt{2T \ln 2/\delta} \tag{3.8}$$

To bound the variance of the selected items, $\sum_{t=1}^{T} s_t(j_t)$, we extend the analysis in (Dani et al., 2008; Wen et al., 2015b) to include the prior covariance term. We thus obtain the following inequality:

$$\sum_{t=1}^{T} s_t(j_t) \leq \sqrt{dnT} \times \sqrt{C \log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)}$$

where $C = \frac{1}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}$. Substituting this into (3.8) completes the proof. □

Note that this theorem assumes that preference vectors θ are smooth according to the given graph structure, implying that the Laplacian term serves as a *correctly specified prior*. Also, observe that since n is large in our case, assumption (a) for the above theorem is reasonable. Assumptions (b) and (c) define the upper and lower bounds on the regularization parameter λ. Similar to Epoch-greedy, transferring information across the graph reduces the regret by a factor dependent on $\mathrm{Tr}(L^{-1})$. Note that compared to epoch-greedy, the regret bound for Thompson sampling has a worse dependence on n, but its $\widetilde{O}(\sqrt{T})$ dependence on T is optimal. If $L = I_{dn}$, we match the $\widetilde{O}(dn\sqrt{T})$ regret bound for a dn-dimensional contextual bandit problem (Abbasi-Yadkori et al., 2011). Note that we have a dependence on d and n similar to the original GOB paper (Cesa-Bianchi et al.,

2013) and that this method performs similarly in practice in terms of regret. However, as will see, our algorithm is much faster.

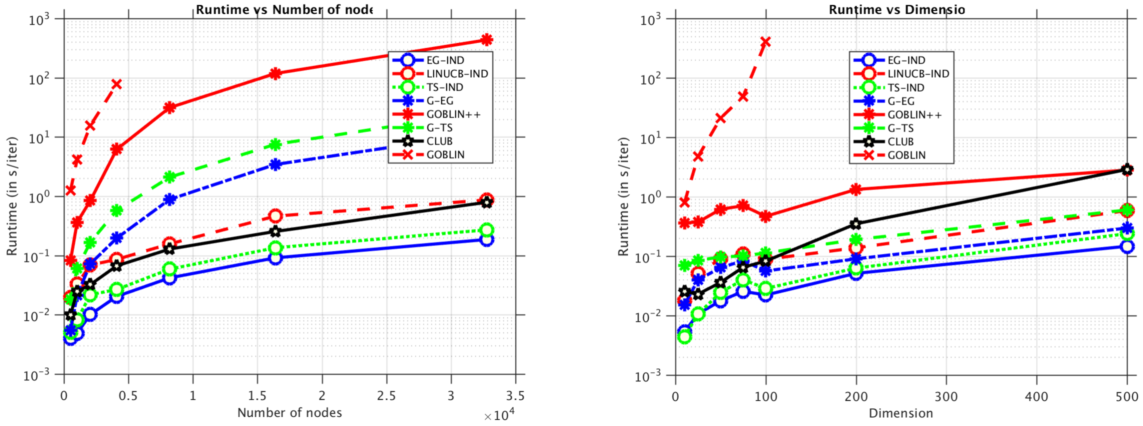
3.4 Experiments

3.4.1 Experimental Setup

Data: We first test the scalability of various algorithms using synthetic data and then evaluate their regret performance on two real datasets. For synthetic data we generate random d-dimensional context vectors and ground-truth user preferences, and generate the ratings according to the linear model. We generated a random Kronecker graph with sparsity 0.005 (which is approximately equal to the sparsity of our real datasets). It is well known that such graphs capture many properties of real-world social networks (Leskovec et al., 2010).

For the real data, we use the Last.fm and Delicious datasets which are available as part of the HetRec 2011 workshop. Last.fm is a music streaming website where each item corresponds to a music artist and the dataset consists of the set of artists each user has listened to. The associated social network consists of 1.8K users (nodes) and 12.7K friendship relations (edges). Delicious is a social bookmarking website, where an item corresponds to a particular URL and the dataset consists of the set of websites bookmarked by each user. Its corresponding social network consists of 1.8K users and 7.6K user-user relations. Similar to (Cesa-Bianchi et al., 2013), we use the set of associated tags to construct the TF-IDF vector for each item and reduce the dimension of these vectors to $d = 25$. An artist (or URL) that a user has listened to (or has bookmarked) is said to be "liked" by the user. In each round, we select a target user uniformly at random and make the set $\mathcal{C}_t$ consist of 25 randomly chosen items such that there is at least 1 item liked by the target user. An item liked by the target user is assigned a reward of 1 whereas other items are assigned a zero reward. We use a total of $T = 50$ thousand recommendation rounds and average our results across 3 runs.

Algorithms: We denote our graph-based epoch-greedy and Thompson sampling algorithms as G-EG and G-TS, respectively. For epoch-greedy, although the analysis suggests that we update the preference estimates only in the exploration rounds, we observed better performance by updating the preference vectors in all rounds (we use this variant in our experiments). We use 10% of the total number of rounds for exploration, and we "exploit" in the remaining rounds. Similar to (Gentile et al., 2014), all hyper-parameters are set using an initial validation set of 5 thousand rounds. The best validation performance was observed for $\lambda = 0.01$ and $\sigma = 1$. To control the amount of exploration for Thompson sampling, we the use posterior reshaping trick (Chapelle and Li, 2011) which reduces the variance of the posterior by a factor of 0.01.



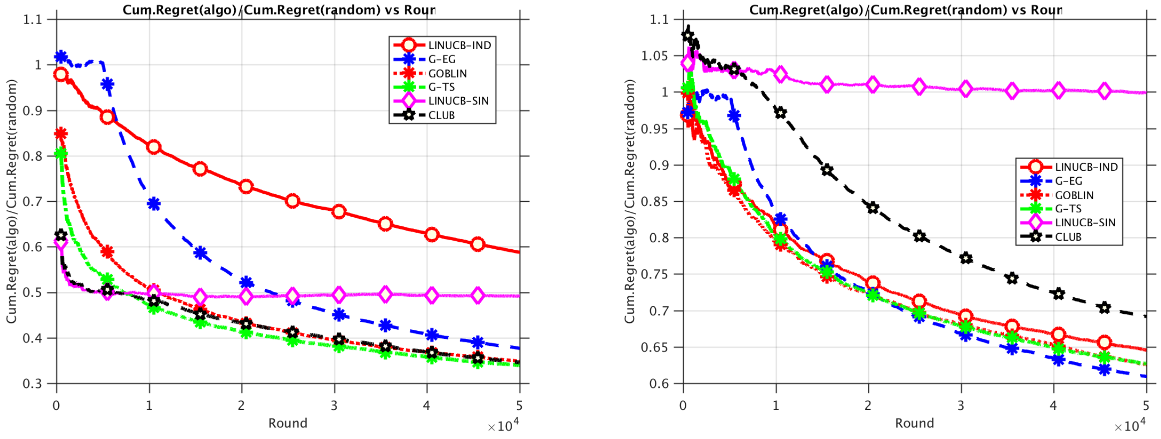
(a)

(b)

Figure 3.1: Synthetic network: Runtime (in seconds/iteration) vs (a) Number of nodes (b) Dimension

Baselines: We consider two variants of graph-based UCB-style algorithms: GOBLIN is the method proposed in the original GOB paper (Cesa-Bianchi et al., 2013) while we use GOBLIN++ to refer to a variant that exploits the fast mean estimation strategy we develop in Section 3.2.3. Similar to (Cesa-Bianchi et al., 2013), for both variants we discount the confidence bound term by a factor of $\alpha = 0.01$.

We also include baselines which ignore the graph structure and make recommendations by solving independent linear contextual bandit problems for each user. We consider 3 variants of this baseline: the LINUCB-IND proposed in (Li et al., 2010), an epoch-greedy variant of this approach (EG-IND), and a Thompson sampling variant (TS-IND). We also compared to a baseline that does no personalization and simply considers a single bandit problem across all users (LINUCB-SIN). Finally, we compared against the state-of-the-art online clustering-based approach proposed in (Gentile et al., 2014), denoted CLUB.



(a) Last.fm

(b) Delicious

Figure 3.2: Regret Minimization

This method starts with a fully connected graph and iteratively deletes edges from the graph based on UCB estimates. CLUB considers each connected component of this graph as a cluster and maintains one preference vector for all the users belonging to a cluster. Following the original work, we make CLUB scalable by generating a random Erdos-Renyi graph $G_{n,p}$ with $p = \frac{3logn}{n}$.[2] In all, we compare our proposed algorithms G-EG and G-TS with 7 reasonable baseline methods.

3.4.2 Results

Scalability: We first evaluate the scalability of the various algorithms with respect to the number of network nodes n. Figure 3.1a shows the runtime in seconds/iteration when we fix $d = 25$ and vary the size of the network from 16 thousand to 33 thousand nodes. Compared to GOBLIN, our proposed GOBLIN++ is more efficient in terms of both time (almost 2 orders of magnitude faster) and memory. Indeed, the existing GOBLIN method runs out of memory even on very small networks and thus we do not plot it for larger networks. Further, our proposed G-EG and G-TS methods scale even more gracefully in the number of nodes and are much faster than GOBLIN++ (although not as fast as the

[2] We reimplemented CLUB. Note that one of the datasets from our experiments was also used in that work and we obtain similar performance to that reported in the original paper.

clustering-based CLUB or methods that ignore the graph).

We next consider scalability with respect to d. Figure 3.1b fixes $n = 1024$ and varies d from 10 to 500. In this figure it is again clear that our proposed GOBLIN++ scales much better than the original GOBLIN algorithm. The EG and TS variants are again even faster, and other key findings from this experiment are (i) it was not faster to ignore the graph and (ii) our proposed G-EG and G-TS methods scale better with d than CLUB.

Regret Minimization: We follow (Gentile et al., 2014) in evaluating recommendation performance by plotting the ratio of cumulative regret incurred by the algorithm divided by the regret incurred by a random selection policy. Figure 3.2a plots this measure for the Last.fm dataset. In this dataset we see that treating the users independently (LINUCB-IND) takes a long time to drive down the regret (we do not plot EG-IND and TS-IND as they had similar performance) while simply aggregating across users (LINUCB-SIN) performs well initially (but eventually stops making progress). We see that the approaches exploiting the graph help learn the user preferences faster than the independent approach and we note that on this dataset our proposed G-TS method performed similar to or slightly better than the state of the art CLUB algorithm.

Figure 3.2b shows performance on the Delicious dataset. On this dataset personalization is more important and we see that the independent method (LINUCB-IND) outperforms the non-personalized (LINUCB-SIN) approach. The need for personalization in this dataset also leads to worse performance of the clustering-based CLUB method, which is outperformed by all methods that model individual users. On this dataset the advantage of using the graph is less clear, as the graph-based methods perform similar to the independent method. Thus, these two experiments suggest that (i) the scalable graph-based methods do no worse than ignoring the graph in cases where the graph is not helpful and (ii) the scalable graph-based methods can do significantly better on datasets where the graph is helpful. Similarly, when user preferences naturally form clusters our proposed methods perform similarly to CLUB, whereas on datasets where individual preferences are important our methods are significantly better.

3.5 Related Work

Social Regularization: Using social information to improve recommendations was first introduced by Ma et al. (Ma et al., 2011). They used matrix factorization to fit existing rating data but constrained a user's latent vector to be similar to their friends in the social network. Other methods based on collaborative filtering followed (Rao et al., 2015; Delporte et al., 2013), but these works assume that we already have rating data available. Thus, these methods do not address the exploration-exploitation trade-off faced by a new RS that we consider.

Several graph-based methods to model dependencies between the users have been explored in the (non-contextual) multi-armed bandit framework (Caron et al., 2012; Mannor and Shamir, 2011; Alon et al., 2014; Maillard and Mannor, 2014), but the GOB model of Cesa-Bianchi et al. (Cesa-Bianchi et al., 2013) is the first to exploit the network between users in the contextual bandit framework. They proposed a UCB-style algorithm and showed that using the graph leads to lower regret from both a theoretical and practical standpoint. However, their algorithm has a time complexity that is quadratic in the number of users. This makes it infeasible for typical RS that have tens of thousands (or even millions) of users.

To scale up the GOB model, several recent works propose to cluster the users and assume that users in the same cluster have the same preferences (Gentile et al., 2014; Nguyen and Lauw, 2014). But this solution loses the ability to model individual users' preferences, and indeed our experiments indicate that in some applications clustering significantly hurts performance. In contrast, we want to scale up the original GOB model that learns more fine-grained information in the form of a preference-vector specific to each user.

Another interesting approach to relax the clustering assumption is to cluster both items and users (Li et al., 2016), but this only applies if we have a fixed set of items. Some works consider item-item similarities to improve recommendations (Valko et al., 2014; Kocák et al., 2014), but this again requires a fixed set of items while we are interested in RS where the set of items may constantly be changing. There has also been work on solving a single bandit problem in a distributed fashion (Korda et al., 2016), but this differs from our approach where we are solving an individual bandit problem on each of the n nodes. Finally, we note that *all* of the existing graph-based works consider relatively small RS datasets ($\sim 1k$ users), while our proposed algorithms can scale to much larger RS.

3.6 Discussion

This work draws a connection between the GOB framework and GMRFs, and uses this to scale up the existing GOB model to much larger graphs. We also proposed and analyzed Thompson sampling and epoch-greedy variants. Our experiments on recommender systems datasets indicate that the Thompson sampling approach in particular is much more scalable than existing GOB methods, obtains theoretically optimal regret, and performs similar to or better than other existing scalable approaches.

In many practical scenarios we do not have an explicit graph structure available. In the appendix, we consider a variant of the GOB model where we use L1-regularization to learn the graph on the fly. Our experiments there show that this approach works similarly to or much better than approaches which use the fixed graph structure. It would be interesting to explore the theoretical properties of this approach.

Chapter 4

Bootstrapping for Bandits

In the previous chapters, we have seen the effectiveness of the linear bandit framework for recommender systems and social networks. However, applications with rich structured data such as images or text require modelling complex non-linear feature-reward mappings. For example, each product (arm) in a recommender system might be associated with an unstructured text review that is useful to infer the arm's expected reward. The common bandit algorithms studied in the literature are not effective or efficient in these complex settings. In this chapter, we propose a bootstrapping based approach to address the exploration-exploitation trade-off for complex non-linear models.

4.1 Introduction

We first highlight the difficulties of the common bandit algorithms in addressing the exploration-exploitation trade-off for non-linear feature-reward mappings. As explained in Chapter 1, the ε-greedy (EG) algorithm is simple to implement and can be directly used with any non-linear function from feature to rewards. However, its performance heavily relies on choosing the right exploration parameter and the strategy for annealing it. The Optimism-in-the-Face-of-Uncertainty (OFU) based strategies are statistically optimal and computationally efficient in the bandit (Auer et al., 2002) and linear bandit (Abbasi-Yadkori et al., 2011) settings. However, in the non-linear setting, we can construct only approximate confidence sets (Filippi et al., 2010b; Li et al., 2017; Zhang et al., 2016; Jun et al., 2017) that result in over-conservative uncertainty estimates (Filippi et al., 2010b) and consequently in worse empirical performance. Similarly, Thompson sampling (TS), which requires drawing a sample from the Bayesian posterior is computationally efficient when we have a closed-form posterior like in the case of Bernoulli or Gaussian rewards. For reward distributions beyond those admitting conjugate priors or for complex non-linear feature-reward mappings, it is not possible to have a closed form posterior or obtain exact samples from it. In these cases, we need to resort to computationally-expensive approximate sampling techniques (Riquelme et al., 2018).

To address the above difficulties, bootstrapping (Efron, 1992) has been used in the bandit (Baransi et al., 2014; Eckles and Kaptein, 2014), contextual bandit (Tang et al., 2015a; McNellis et al., 2017) and deep reinforcement learning (Osband and Van Roy, 2015; Osband et al., 2016) settings. This previous work uses the classic *non-parametric bootstrapping* procedure (detailed in Section 4.3.1) as an approximation to TS. As opposed to maintaining the entire posterior distribution for TS, bootstrapping requires computing only point-estimates (such as the maximum likelihood estimator). Bootstrapping thus has two major advantages over other existing strategies: (i) Unlike OFU and TS, it is simple to implement and does not require designing problem-specific confidence sets or efficient sampling algorithms. (ii) Unlike EG, it is not overly sensitive to hyper-parameter tuning.

In spite of its advantages and good empirical performance, bootstrapping for bandits is not well understood theoretically, even under special settings such as the K-armed bandit problem with Bernoulli or Gaussian rewards. Indeed, to the best of our knowledge, McNellis et al. (2017) is the only work that attempts to theoretically analyze the non-parametric bootstrapping (referred to as NPB) procedure. For the bandit setting with Bernoulli rewards and a Beta prior (henceforth referred to as the Bernoulli bandit setting), they prove that both TS and NPB will take similar actions as the number of rounds increases. However, they do not provide any explicit regret bounds for NPB.

In this chapter, we first show that the NPB procedure used in the previous work can be provably inefficient in the Bernoulli bandit setting (Section 4.3.2). In particular, we establish a near-linear lower bound on the incurred regret. In Section 4.3.3, we show that NPB with an appropriate amount of *forced exploration*, typically done in practice in (McNellis et al., 2017; Tang et al., 2015a), can result in a sub-linear upper bound on the regret, which nevertheless remains suboptimal. As an alternative to NPB, we propose the *weighted bootstrapping* (abbreviated as WB) procedure. For Bernoulli (or more generally categorical) rewards, we prove that WB with multiplicative exponential weights is mathematically equivalent to TS and thus results in near-optimal regret. Note that this connection was made independently in the earlier work of Osband and Van Roy (2015). However, unlike us, they do not experimentally evaluate the effectiveness of the weighted bootstrapping algorithm in the bandit setting. Moreover, we also show that for Gaussian rewards, WB with additive Gaussian weights is equivalent to TS with an uninformative prior and also attains near-optimal regret.

In Section 4.5, we first experimentally compare the performance of WB, NPB and TS in the the multi-armed bandit setting. We show that for several reward distributions on $[0, 1]$, WB (and NPB) outperforms a modified TS procedure proposed in (Agrawal and Goyal, 2013b). In the contextual bandit setting, we experimentally evaluate the bootstrapping procedures with several parametric models and real-world datasets. In this setting, we give practical guidelines for making computationally efficient updates to the model and for initializing the bootstrapping procedure.

For computational efficiency, prior work (Eckles and Kaptein, 2014; McNellis et al., 2017; Tang et al., 2015a) approximated the bootstrapping procedure by making incremental updates to an ensemble of models. Such an approximation requires additional hyper-parameter tuning, such as choosing the size of the ensemble; or problem-specific heuristics, for example McNellis et al. (2017) use a lazy update procedure specific to decision trees. In contrast, we find that with appropriate stochastic optimization, bootstrapping (without any approximation) for parametric models is computationally efficient and simple to implement.

Another design decision involves the initialization of the bootstrapping procedure. Prior work (Eckles and Kaptein, 2014; McNellis et al., 2017; Tang et al., 2015a) uses forced exploration at the beginning of bootstrapping. For this, the work in (Eckles and Kaptein, 2014) uses pseudo-examples in order to simulate a Beta prior before starting the NPB procedure for the MAB problem. In the contextual bandit setting, both McNellis et al. (2017); Tang et al. (2015a) initialize the bootstrapping procedure by pulling each arm a minimum number of times or by generating a "sufficient" number of pseudo-examples. It is not clear how to generate pseudo-examples in this setting; for example, McNellis et al. (2017) recommend using features of the context vector in the first round for generating the pseudo-examples. However, we observe that such a procedure results in under-exploration and worse performance.

In Section 4.5, we propose a simple method for generating such examples. We empirically validate that both NPB and WB in conjunction with this initialization results in consistently good performance. Our contributions result in a simple and efficient implementation of the bootstrapping procedure that has theoretical guarantees in the simple Bernoulli and Gaussian MAB setting.

4.2 Background

In this section, we give the necessary background on bootstrapping and then explain its adaptation to bandits in Section 4.2.2.

4.2.1 Bootstrapping

Bootstrapping is typically used to obtain uncertainty estimates for a model fit to data. The general bootstrapping procedure consists of two steps: (i) Formulate a *bootstrapping log-likelihood* function $\widetilde{\mathcal{L}}(\theta, Z)$ by injecting stochasticity into the log-likelihood function $\mathcal{L}(\cdot)$ via the random variable Z such that $\mathbb{E}_Z\left[\widetilde{\mathcal{L}}(\theta, Z)\right] = \mathcal{L}(\theta)$. (ii) Given $Z = z$, generate a *bootstrap sample* $\widetilde{\theta}$ as: $\widetilde{\theta} \in \arg\max_\theta \widetilde{\mathcal{L}}(\theta, z)$. In the offline setting (Friedman et al., 2001), these steps are repeated B (usually $B = 10^4$) times to obtain the set $\{\widetilde{\theta}^1, \widetilde{\theta}^2, \ldots \widetilde{\theta}^B\}$. The variance of these samples is then used to estimate the uncertainty in the model parameters $\widehat{\theta}$. Unlike a Bayesian approach that requires characterizing the entire posterior distribution in order to compute uncertainty estimates, bootstrapping only requires computing point-estimates (maximizers of the bootstrapped log-likelihood functions). In Sections 4.3 and 4.4, we discuss two specific bootstrapping procedures.

4.2.2 Bootstrapping for Bandits

Algorithm 4 Bootstrapping for contextual bandits

```
Input: $K$ arms, Model class $m$
Initialize history: $\forall j \in [K]$, $\mathcal{D}_j = \{\}$
for $t = 1$ to $T$ do
    Observe context vector $\mathbf{x}_t$
    For all $j$, compute bootstrap sample $\widetilde{\theta}_j$ (According to Sections 4.3 and 4.4)
    Select arm: $j_t = \arg\max_{j \in [K]} m(\mathbf{x}_t, \widetilde{\theta}_j)$
    Observe reward $r_t$
    Update history: $\mathcal{D}_{j_t} = \mathcal{D}_{j_t} \cup \{\mathbf{x}_t, r_t\}$
```

In the bandit setting, the work in (Eckles and Kaptein, 2014; Tang et al., 2015a; McNellis et al., 2017) uses bootstrapping as an approximation to Thompson sampling (TS). The basic idea is to compute *one bootstrap sample* and treat it as a sample from an underlying posterior distribution in order to emulate TS. In Algorithm 4, we describe the procedure for the contextual bandit setting. At every round t, the set $\mathcal{D}_j$ consists of the features and observations obtained on pulling arm j in the previous rounds. The algorithm (in line 5) uses the set $\mathcal{D}_j$ to compute a bootstrap sample $\widetilde{\theta}_j$ for each arm j. Given the bootstrap sample for each arm, the algorithm (similar to TS) selects the arm j_t maximizing the reward conditioned on this bootstrap sample (line 6). After obtaining the observation (line 7), the algorithm updates the set of observations for the selected arm (line 8). In the subsequent sections, we instantiate the procedures for generating the bootstrap sample $\widetilde{\theta}_j$ and analyze the performance of the algorithm in these settings.

4.3 Non-parametric Bootstrapping

We first describe the non-parametric bootstrapping (NPB) procedure in Section 4.3.1. We show that NPB used in conjunction with Algorithm 4 (Eckles and Kaptein, 2014) can be provably inefficient and establish a near-linear lower bound on the regret incurred by it in the Bernoulli bandit setting (Section 4.3.2). In Section 4.3.3, we show that NPB with an appropriate amount of forced exploration can result in an $O(T^{2/3})$ regret in this setting.

4.3.1 Procedure

In order to construct the bootstrap sample $\widetilde{\theta}_j$ in Algorithm 4, we first construct a new dataset $\widetilde{\mathcal{D}}_j$ by *sampling with replacement*, $|\mathcal{D}_j|$ points from the set $\mathcal{D}_j$. The bootstrapped log-likelihood is equal to the log-likelihood of observing $\widetilde{\mathcal{D}}_j$. Formally,

$$\widetilde{\mathcal{L}}(\theta) = \sum_{i \in \widetilde{\mathcal{D}}_j} \log\left[\mathcal{P}(y_i | x_i, \theta)\right] \tag{4.1}$$

The bootstrap sample is computed as $\widetilde{\theta}_j \in \arg\max_\theta \widetilde{\mathcal{L}}(\theta)$. Observe that the sampling with replacement procedure is the source of randomness for the bootstrapping and that $\mathbb{E}[\widetilde{\mathcal{D}}_j] = \mathcal{D}_j$.

For the special case of Bernoulli rewards without features, a common practice is to use *Laplace smoothing* where we generate positive (1) or negative (0) *pseudo-examples* to be used in addition to the observed rewards. Laplace smoothing is associated with two non-negative integers α_0, β_0, where α_0 (and β_0) is the *pseudo-count*, equal to the number of positive (or negative) pseudo-examples. These pseudo-counts are used to "simulate" the prior distribution $Beta(\alpha_0, \beta_0)$. For the NPB procedure with Bernoulli rewards, generating $\widetilde{\theta}_j$ is equivalent to sampling from a Binomial distribution $Bin(n, p)$ where $n = |\mathcal{D}_j|$ and the success probability p is equal to the fraction of positive observations in $\mathcal{D}_j$. Formally, if the number of positive observations in $\mathcal{D}_j$ is equal to α, then

$$A \sim \text{Bino}\left(n + \alpha_0 + \beta_0, \frac{\alpha_0 + \alpha}{n + \alpha_0 + \beta_0}\right) \quad \text{and} \quad \widetilde{\theta}_j = \frac{A}{n + \alpha_0 + \beta_0} \tag{4.2}$$

4.3.2 Inefficiency of Non-Parametric Bootstrapping

In this subsection, we formally show that Algorithm 4 used with NPB might lead to an $\Omega(T^\gamma)$ regret with γ arbitrarily close to 1. Specifically, we consider a simple 2-arm bandit setting, where at each round t, the reward of arm 1 is independently drawn from a Bernoulli distribution with mean $\mu_1 = 1/2$, and the reward of arm 2 is deterministic and equal to 1/4. Furthermore, we assume that the agent knows the deterministic reward of arm 2, but not the mean reward for arm 1. Notice that this case is simpler than the standard two-arm Bernoulli bandit setting, in the sense that the agent also knows the reward of arm 2. Observe that if $\widetilde{\theta}_1$ is a bootstrap sample for arm 1 (obtained according to equation 4.2), then the arm 1 is selected if $\widetilde{\theta}_1 \geq 1/4$. Under this setting, we prove the following lower bound:

Theorem 6. *If the NPB procedure is used in the above-described case with pseudo-counts* $(\alpha_0, \beta_0) = (1, 1)$ *for arm* 1*, then for any* $\gamma \in (0, 1)$ *and any* $T \geq \exp\left[\frac{2}{\gamma}\exp\left(\frac{80}{\gamma}\right)\right]$*, we obtain*

$$\mathbb{E}[R(T)] > \frac{T^{1-\gamma}}{32} = \Omega(T^{1-\gamma}).$$

Proof. Please refer to Appendix C.1 for the detailed proof of Theorem 6. It is proved based on a binomial tail bound (Proposition 2) and uses the following observation: under a "bad history", where at round τ NPB has pulled arm 1 for m times, but all of these m pulls have resulted in a reward 0, NPB will pull arm 1 with probability less than $\exp(-m\log(m)/20)$ (Lemma 21). Hence, the number of times NPB will pull the suboptimal arm 2 before it pulls arm 1 again or reach the end of the T time steps follows a "truncated geometric distribution", whose expected value is bounded in Lemma 22. Based on Lemma 22, and the fact that the probability of this bad history is 2^{-m}, we have $\mathbb{E}\left[R(T)\right] \geq 2^{-(m+3)} \min\left\{\exp\left(m\log(m)/20\right), T/4\right\}$ in Lemma 23. Theorem 6 is proved by setting $m = \lceil \gamma \log(T)/2 \rceil$. □

Theorem 6 shows that in the Bernoulli bandit setting, when T is large enough, the NPB procedure used in previous work (Eckles and Kaptein, 2014; Tang et al., 2015a; McNellis et al., 2017) can incur an expected cumulative regret arbitrarily close to a linear regret in the order of T. It is straightforward to prove a variant of this lower bound with any constant (in terms of T) number of pseudo-examples. Next, we show that NPB with forced exploration that depends on the horizon T can result in sub-linear regret.

4.3.3 Forced Exploration

In this subsection, we show that NPB, when coupled with an appropriate amount of forced exploration, can result in sub-linear regret in the Bernoulli bandit setting. In order to *force* exploration, we pull each arm m times before starting Algorithm 4. Note that a similar procedure for forcing exploration is used in the contextual bandit setting in (McNellis et al., 2017; Tang et al., 2015a). The following theorem shows that for an appropriate value of m, this strategy can result in an $O(T^{2/3})$ upper bound on the regret.

Theorem 7. *In any 2-armed bandit setting, if each arm is initially pulled* $m = \left\lceil \left(\frac{16\log T}{T}\right)^{\frac{1}{3}} \right\rceil$ *times before starting Algorithm 4, then*

$$\mathbb{E}[R(T)] = O(T^{2/3}) \, .$$

Proof. The claim is proved in appendix C.2 based on the following observation: If the gap of the suboptimal arm is large, the prescribed m steps are sufficient to guarantee that the bootstrap sample of the optimal arm is higher than that of the suboptimal arm with a high probability at any round t. On the other hand, if the gap of the suboptimal arm is small, no algorithm can have high regret. □

Although forced exploration is able to remedy the NPB procedure, we can prove only a sub-optimal regret bound for this strategy. In the next section, we consider a simple weighted bootstrapping approach and show that it can lead to a near-optimal regret bound in the Bernoulli bandit setting.

4.4 Weighted Bootstrapping

In this section, we propose weighted bootstrapping (WB) as an alternative to the non-parametric bootstrap. We first describe the weighted bootstrapping procedure in Section 4.4.1. For the bandit setting with Bernoulli rewards, we show the mathematical equivalence between WB and TS, hence proving that WB attains near-optimal regret (Section 4.4.2).

4.4.1 Procedure

In order to formulate the bootstrapped log-likelihood, we use a *random transformation* of the labels in the corresponding log-likelihood function. First, consider the case of Bernoulli observations where the labels $y_i \in \{0, 1\}$. In this case, the log-likelihood function is given by:

$$\mathcal{L}(\theta) = \sum_{i \in \mathcal{D}_j} y_i \log\left(g\left(\langle \mathbf{x}_i, \theta \rangle\right)\right) + (1 - y_i) \log\left(1 - g\left(\langle \mathbf{x}_i, \theta \rangle\right)\right)$$

where the function $g(\cdot)$ is the inverse-link function. For each observation i, we sample a random weight w_i from an exponential distribution, specifically, for all $i \in \mathcal{D}_j$, $w_i \sim Exp(1)$. We use the following transformation of the labels: $y_i :\rightarrow w_i \cdot y_i$ and $(1 - y_i) :\rightarrow w_i \cdot (1 - y_i)$. Since we transform the labels by multiplying them with exponential weights, we refer to this case as *WB with multiplicative exponential weights.* Given this transformation, the bootstrapped log-likelihood function is defined as:

$$\widetilde{\mathcal{L}}(\theta) = \sum_{i \in \mathcal{D}_j} w_i \underbrace{\left[y_i \log\left(g\left(\langle \mathbf{x}_i, \theta \rangle\right)\right) + (1 - y_i) \log\left(1 - g\left(\langle \mathbf{x}_i, \theta \rangle\right)\right)\right]}_{\ell_i(\theta)} = \sum_{i \in \mathcal{D}_j} w_i \cdot l_i(\theta) \tag{4.3}$$

Here ℓ_i is the log-likelihood of observing point i. As before, the bootstrap sample is computed as: $\widetilde{\theta}_j \in \arg\max_\theta \widetilde{\mathcal{L}}(\theta)$.

In WB, the randomness for bootstrapping is induced by the weights w and that $\mathbb{E}_w[\widetilde{\mathcal{L}}(\theta)] = \mathcal{L}(\theta)$. Let us consider a special case, in the absence of features, when $g\left(\langle \mathbf{x}_i, \theta \rangle\right) = \theta$ for all

i. Assuming α_0 positive and β_0 negative pseudo-counts and denoting $n = |\mathcal{D}_j|$, we obtain the following closed-form expression for computing the bootstrap sample:

$$\widetilde{\theta} = \frac{\sum_{i=1}^{n} [w_i \cdot y_i] + \sum_{i=1}^{\alpha_0} [w_i]}{\sum_{i=1}^{n+\alpha_0+\beta_0} w_i} \tag{4.4}$$

Observe that the above transformation procedure extends the domain for the labels from values in $\{0, 1\}$ to those in $\mathbb{R}$ and does not result in a valid probability mass function. However, this transformation has the following advantages: (i) Using equation 4.3, we can interpret $\widetilde{\mathcal{L}}(\theta)$ as a random re-weighting (by the weights w_i) of the observations. This formulation is equivalent to the weighted likelihood bootstrapping procedure proposed and proven to be asymptotically consistent in the offline case in (Newton and Raftery, 1994). (ii) From an implementation perspective, computing $\widetilde{\theta}_j$ involves solving a weighted maximum likelihood estimation problem. It thus has the same computational complexity as NPB and can be solved by using black-box optimization routines. (iii) In the next section, we show that using WB with multiplicative exponential weights has good theoretical properties in the bandit setting. Furthermore, such a procedure of randomly transforming the labels lends itself naturally to the Gaussian case and in Appendix C.3.2, we show that WB with an additive transformation using Gaussian weights is equivalent to TS.

4.4.2 Equivalence to Thompson sampling

We now analyze the theoretical performance of WB in the Bernoulli bandit setting. In the following proposition proved in appendix C.3.1, we show that WB with multiplicative exponential weights is equivalent to TS.

Proposition 1. *If the rewards $y_i \sim Ber(\theta^*)$, then weighted bootstrapping using the estimator in equation 4.4 results in $\widetilde{\theta}_j \sim Beta(\alpha + \alpha_0, \beta + \beta_0)$, where α and β is the number of positive and negative observations respectively; α_0 and β_0 are the positive and negative pseudo-counts. In this case, WB is equivalent to Thompson sampling under the $Beta(\alpha_0, \beta_0)$ prior.*

Since WB is mathematically equivalent to TS, the bounds in (Agrawal and Goyal, 2013a) imply near-optimal regret for WB in the Bernoulli bandit setting.

In Appendix C.3.1, we show that this equivalence extends to the more general categorical (with C categories) reward distribution i.e. for $y_i \in \{1, \ldots C\}$. In appendix C.3.2, we prove that for Gaussian rewards, WB with additive Gaussian weights, i.e. $w_i \sim N(0,1)$ and using the additive transformation $y_i :\to y_i + w_i$, is equivalent to TS under an uninformative $N(0,\infty)$ prior. Furthermore, this equivalence holds even in the presence of features, i.e. in the linear bandit case. Using the results in (Agrawal and Goyal, 2013b), this implies that for Gaussian rewards, WB with additive Gaussian weights achieves near-optimal regret.

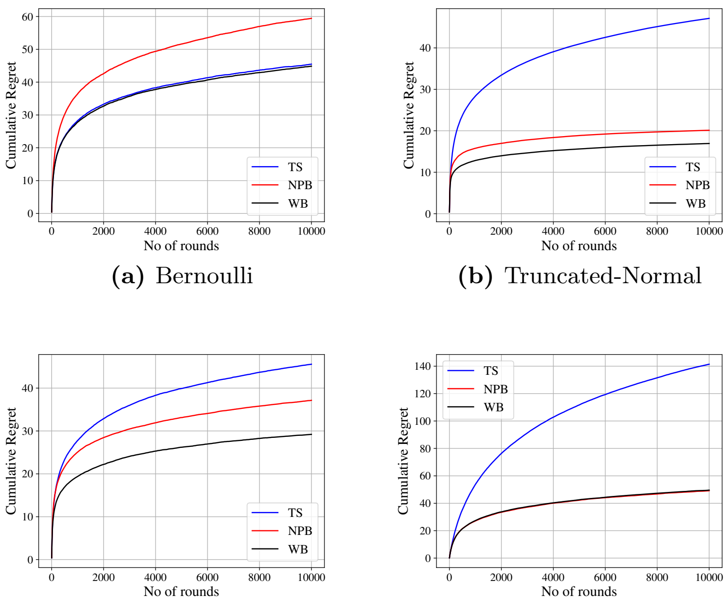
4.5 Experiments

In Section 4.5.1, we first compare the empirical performance of bootstrapping and Thompson sampling in the bandit setting. In Section 4.5.2, we describe the experimental setup for the contextual bandit setting and compare the performance of different algorithms under different feature-reward mappings.

4.5.1 Bandit setting

We consider $K = 10$ arms (refer to Appendix C.4 for results with other values of K), a horizon of $T = 10^4$ rounds and average our results across 10^3 runs. We perform experiments for four different reward distributions - Bernoulli, Truncated Normal, Beta and the Triangular distribution (Kotz and Van Dorp, 2004), all bounded on the $[0,1]$ interval. In each run and for each arm j, we choose the expected reward μ_j (mean of the corresponding distribution) to be a uniformly distributed random number in $[0,1]$. For the Truncated-Normal distribution, we choose the standard deviation to be equal to 10^{-4} (we also experimented with other values of the standard deviation and observed similar trends), whereas for the Beta distribution, the shape parameters of arm j are chosen to be $\alpha = \mu_j$ and $\beta = 1 - \mu_j$. We use the $Beta(1,1)$ prior for TS. In order to use TS on distributions other than Bernoulli, we follow the procedure proposed in (Agrawal and Goyal, 2013a): for a reward in $[0,1]$ we flip a coin with the probability of obtaining 1 equal to the reward, resulting in a binary "pseudo-reward". This pseudo-reward is then used to update the Beta posterior as in the Bernoulli case. For NPB and WB, we use the estimators in equations 4.2 and 4.4 respectively. For both of these, we use the pseudo-counts $\alpha_0 = \beta_0 = 1$.

In the Bernoulli case, NPB obtains a higher regret as compared to both TS and WB which are equivalent. For the other distributions, we observe that both WB and NPB (with WB resulting in consistently better performance) obtain lower cumulative regret than the modified TS procedure. This shows that for distributions that do not admit a conjugate



(a) Bernoulli

(b) Truncated-Normal

(c) Beta

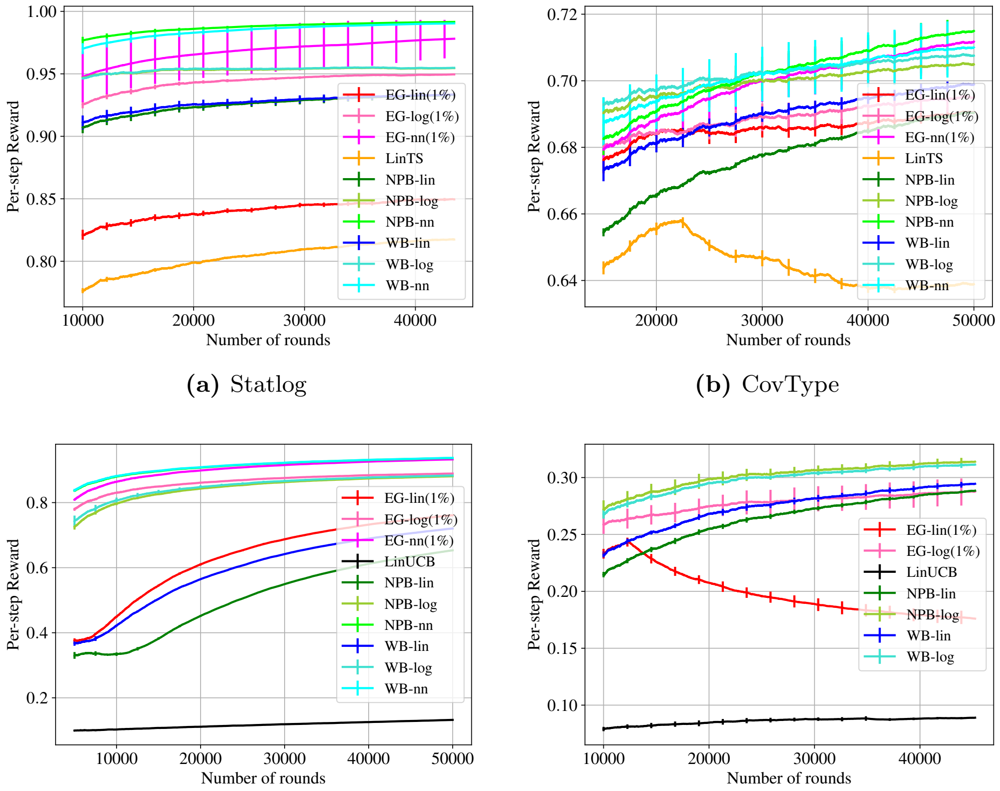
(d) Triangular

Figure 4.1: Cumulative Regret vs Number of rounds for TS, NPB and WB in a bandit setting with $K = 10$ arms for (a) Bernoulli (b) Truncated-Normal (c) Beta (d) Triangular reward distributions bounded on the $[0, 1]$ interval. WB results in the best performance in each these experiments.

prior, WB (and NPB) can be directly used and results in good empirical performance as compared to making modifications to the TS procedure.

4.5.2 Contextual bandit setting

We adopt the one-versus-all multi-class classification setting for evaluating contextual bandits (Agarwal et al., 2014; McNellis et al., 2017; Riquelme et al., 2018). In this setting, arm k corresponds to class $k \in [K]$. At time t, the algorithm observes context vector $x_t \in \mathbb{R}^{d \times 1}$ and then pulls an arm. It receives a reward of one if the pulled arm corresponds to the correct class, and zero otherwise. Each arm maintains an independent set of statistics that map x_t to the observed binary reward. We use four multi-class datasets from (Riquelme et al., 2018): Statlog ($d = 9, K = 7$), CovType ($d = 54, K = 7$), MNIST($d = 784$, $K = 10$) and Adult ($d = 94, K = 14$). We preprocess these datasets by adding a bias term and



(a) Statlog

(b) CovType

(c) MNIST

(d) Adult

Figure 4.2: Expected per-step reward vs Number of Rounds for (a) Statlog (b) CovType (c) MNIST (d) Adult datasets. Bootstrapping approaches with linear regression consistently perform better than LinUCB, LinTS and Linear EG. Whereas the performance of NPB and WB with non-linear models is close to or better than that of the corresponding non-linear EG models.

standardizing the feature vectors.

The time horizon is $T = 50000$ steps and our results are averaged over 5 runs. We compare the performance of non-parametric bootstrapping (NPB) and weighted bootstrapping (WB) to LinUCB (Abbasi-Yadkori et al., 2011), linear TS (LinTS) (Agrawal and Goyal, 2013b), ε-greedy (EG) (Langford and Zhang, 2008). We also implemented GLM-UCB (Li et al., 2017). GLM-UCB consistently over-explored and had worse performance that EG or LinTS. Therefore, we do not report these results. We run EG, NPB and WB with three classes of models: linear regression (suffix *lin* in plots), logistic regression (suffix *log* in plots), and a single hidden-layer fully-connected neural network (suffix *nn* in plots); with 10 hidden neurons for the Statlog, CovType and Adult datasets and with 100 hidden neurons for the MNIST dataset. We experimented with different exploration schedules in EG. The best performing schedule across all three datasets was $\varepsilon_t = b/t$, where b is set to achieve 1% exploration in T steps. Note that this gives EG an unfair advantage, since such tuning cannot be done for a new online problem.

For EG and the bootstrapping approaches, we solve the maximum likelihood estimation (MLE) problem at each step using stochastic optimization, which is warm-started with the solution from the previous step. For linear and logistic regression, we optimize until the error drops below 10^{-3}. For neural networks, we make one pass over the dataset at each step. To ensure that our results do not depend on the specific choice of optimization, we use publicly available optimization libraries. For linear and logistic regression, we use scikit-learn (Buitinck et al., 2013) with stochastic optimization and its default options. For neural networks, we use the Keras library (Chollet, 2015) with the ReLU non-linearity for the hidden layer and a sigmoid output layer, along with SGD and its default configuration. This is in contrast to (McNellis et al., 2017; Tang et al., 2015a), who approximate bootstrapping by maintaining an ensemble of models. Our preliminary experiments suggested that our procedure yields similar or better solutions than the method proposed in (McNellis et al., 2017) with a better run time, and yields better solutions than (Tang et al., 2015a) without any hyper-parameter tuning. We defer these comparison results to Appendix C.4.2. For both NPB and WB, we use $\log(T)$(≈ 4 for the datasets considered) pseudo-examples in all our experiments. For the features corresponding to the pseudo-examples, we independently sample each dimension from a standard normal distribution. We generate equal number of positive (with label 1) and negative pseudo-examples (with label 0). We find this that choice results in good empirical performance across model-classes and datasets.

Since we compare multiple bandit algorithms and model-classes simultaneously, we use the expected per-step reward in T steps, $\mathbb{E}\sum_{t=1}^{T} r_t/T$, as our performance metric. The expected per-step reward in all three datasets is reported in 4.2. We observe the following trends: first, both linear methods, LinTS and LinUCB, perform the worst[1]. Second, linear variants of both NPB and WB perform comparably to linear EG on the Statlog, CovType and MNIST datasets. On the Adult dataset in fig. 4.2d, EG does not explore enough for

[1] To avoid clutter in the plots, we only plot the better performing method among LinTS and LinUCB.

Dataset	EG-log	EG-nn	NPB-log	NPB-nn	WB-log	WB-nn
Statlog	0.15	0.11	0.035	0.093	0.032	0.10
CovType	0.30	0.19	0.062	0.14	0.061	0.14
MNIST	1.8	2.98	0.29	0.77	0.66	0.52
Adult	0.49	-	0.72	-	0.50	-

Table 4.1: Runtime in seconds/round for non-linear variants of EG, NPB and WB.

the relatively larger number of arms. In contrast, both WB and NPB explore enough and perform well. Third, non-linear variants of EG, NPB and WB typically perform better than their linear counterparts. The most expressive generalization model, the neural network, outperforms logistic regression on the Statlog, CovType and MNIST datasets. This shows that even for relatively simple datasets, like Statlog and CovType, a more expressive non-linear model can lead to better performance. This effect is more pronounced for the MNIST dataset in figure 4.2c. Finally, on the Adult dataset, the neural network, which we do not plot, performs the worst. To investigated this further, we trained a neural network offline for each arm with all available data. Even in this case, we observed that the neural network performs worse than linear regression. We conclude that the poor performance is because of the lack of training data for all the arms, and not because of the lack of exploration.

In order to showcase the computational efficiency of the bootstrapping approaches, we present the run-times for the non-linear variants of EG, NPB and WB for the four datasets in Table 4.1.

4.6 Discussion

We showed that the commonly used non-parametric bootstrapping procedure can be provably inefficient. As an alternative, we proposed the weighted bootstrapping procedure, special cases of which become equivalent to TS for common reward distributions such as Bernoulli and Gaussian. On the empirical side, we showed that the WB procedure has better performance than a modified TS scheme for several bounded distributions in the bandit setting. In the contextual bandit setting, we provided guidelines to make bootstrapping simple and efficient to implement and showed that non-linear versions of bootstrapping have good empirical performance. Our work raises several open questions: does bootstrapping result in near-optimal regret for generalized linear models? Under what assumptions or modifications can NPB be shown to have good performance? On the empirical side, evaluating bootstrapping across multiple datasets and comparing it against TS with approximate sampling is an important future direction.

Chapter 5

Discussion

In this thesis, we used the framework of structured bandits to address decision-making under uncertainty for problems arising primarily in social networks and recommender systems. In Chapter 2, we addressed the influence maximization problem in social networks. We shed light on the inherent complexity of the IM bandits problem. Furthermore, we developed a learning framework that is independent of the underlying model of diffusion. Our framework ensures that the diffusion model can be learnt efficiently, both from a statistical and computational point of view. We believe that our framework addresses some key challenges in making influence maximization practical in the real-world.

Similarly, in Chapter 3, we addressed the problem of using additional network information to make better recommendations in the contextual bandit setting. Our proposed algorithms are able to scale to large-scale real-world problems and have regret guarantees under reasonable assumptions. We hope that this work provides practical ideas and theoretical insight to better incorporate social information for addressing the cold-start problem in recommender systems.

Finally, in Chapter 4, we attempted to relax the linear bandit assumption made in the earlier chapters by turning to the classic notion of bootstrapping. We showed that the common idea of non-parametric bootstrapping for bandit problems can be provably inefficient and devised a weighted bootstrapping algorithm that has guarantees in simple yet realistic settings. In our opinion, it is extremely important for the community to be able to efficiently construct uncertainty estimates and develop principled ways of inducing exploration with complex machine learning models. We hope that these techniques will have wide-spread applications to not only bandits, but also to active learning and reinforcement learning.

All in all, we strongly believe that by making the right structural assumptions, it is possible to devise scalable data-driven approaches that not only address important practical problems, but are also principled from a theoretical point of view. To that end, we discuss some future extensions of the work presented in this thesis. These are presented on a chapter by chapter basis as follows:

- **Chapter 2**
  - **Alternative diffusion models**: We have considered only discrete-time progressive diffusion models in our bandit framework. It will be interesting to quantify the exploration-exploitation trade-off for continuous-time diffusion models (Gomez Rodriguez et al., 2012; Du et al., 2013) or non-progressive models where activated nodes can become inactive again (Lou et al., 2014).
  - **Adaptive IM bandits**: It is important to study the effects of interventions as the diffusion is taking place and adapt the seed-selection to it (Vaswani and Lakshmanan, 2016; Han et al., 2018). Being able to model such effects brings the framework of IM bandits closer to reinforcement learning since an intervention changes the "state" of the diffusion. Such a setting has been recently studied in the context of point processes (Upadhyay et al., 2018).
  - **Contextual bandits and feature construction**: The proposed IMB framework allows for any set of features that are predictive of the influence probabilities. It can thus be used to model product-specific features (for instance, topics (Aslay et al., 2014)) as context vectors in a contextual bandit framework and also allow the probabilities to change across rounds (Bao et al., 2016).

    Furthermore, one can exploit the recent advances in graph embeddings (Grover and Leskovec, 2016; Kipf and Welling, 2016; Hartford et al., 2018) for constructing better features that may results in a lower regret in practice. It might be possible to learn these embeddings in an end-to-end manner for the precise task of influence maximization. For the model-independent IMB framework in Section 2.4, if we can obtain reasonable features for predicting the reachability probabilities, we do not even need to know the graph structure. This framework can also be used to model dynamic graphs changing across the IM attempts.

- **Chapter 3**

  - **Theory for learning the graph**: In Appendix B, we showed that it is possible to learn the graph on-the-fly within the bandit framework. An important direction for future work would be to provide regret guarantees for this scheme that learns both the user preferences and graph structure simultaneously.
  - **Semi supervised and Multi-task learning**: The HOB framework can be used beyond content-based recommendation. In particular, it could be useful in the graph-based semi-supervised learning (Zhu, 2005) or multi-task learning (Evgeniou et al., 2005) in the bandit setting.
  - **Prior misspecification**: Our regret guarantees in Chapter 3 assume that the graph acts as a correctly specified prior, in that the user preferences are smooth according to the given graph. An important extension of this work would be to quantify the effect of prior misspecification on the cumulative regret (for example, using the techniques in (Liu and Li, 2016)).
  - **User selection**: We assume that the target user (to whom recommendations are made) is chosen randomly in every round. Given a set of prospective users to which recommendations can be made, is it possible to learn the user preferences faster by considering interactions between this set of target users? Alternatively, if some users are "available" for recommendation more often than others, is it possible to use the graph to learn the preferences for all the users uniformly well?
  - **Combining with collaborative filtering**: An important practical extension would be to have a systematic procedure for combining our framework for content-based recommendation with traditional collaborative filtering approaches, for example using the approaches in (Rao et al., 2015; Gentile et al., 2017).

- **Chapter 4**

  - **Scalability**: The computational complexity for the proposed bootstrapping approaches is linear in the number of observations. An interesting future direction would be to exploit techniques such as Bag-of-Little-Bootstraps (Kleiner et al., 2014) in order to approximate the non-parametric bootstrapping procedure, use influence functions (Koh and Liang, 2017) in order to quickly estimate the bootstrap estimator or use techniques for the online estimation of the MLE and still provide regret guarantees (Jun et al., 2017).

  - **Provable algorithms for generalized linear models**: We showed that the proposed bootstrapping approaches have provable regret guarantees for the Bernoulli MAB and linear bandit problems. An important future contribution would be to show that the bootstrapping approaches can lead to provably better regret as compared to the UCB or Thompson sampling approaches in the context of generalized linear models.
  - **Uncertainty estimates in other applications**: We have used bootstrapping for estimating the uncertainty in order to trade-off exploration and exploitation in the bandit setting. It will be interesting to use the proposed bootstrapping approaches and their corresponding uncertainty estimates for active learning or reinforcement learning (Osband et al., 2016).

- **Further extensions**: Finally, both the applications in Chapters 2 and 3 can benefit from using more expressive non-linear models and using the bootstrapping approach for trading off exploration and exploitation. We leave these extensions as important practical directions to explore.

Other important future work includes studying the pure-exploration (Bubeck et al., 2009; Soare et al., 2014; Chen et al., 2014) setting and be able to model safety or other constraints (Wu et al., 2016; Kazerouni et al., 2017) in our bandit frameworks.

Bibliography

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. → pages 4, 7, 8, 23, 52, 59, 70, 111, 112, 127, 128, 129, 144, 145

Marc Abeille and Alessandro Lazaric. Linear thompson sampling revisited. *arXiv preprint arXiv:1611.06534*, 2016. → page 8

Alekh Agarwal, Daniel J. Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1638–1646, 2014. URL http://jmlr.org/proceedings/papers/v32/agarwalb14.html. → page 69

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, et al. A multiworld testing decision service. arxiv preprint. 2016. → page 1

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceeding of the 25th Annual Conference on Learning Theory*, pages 39.1–39.26, 2012a. → page 8

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. *arXiv preprint arXiv:1209.3352*, 2012b. → pages 8, 44, 51, 52, 138, 140, 145, 146

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, 2013a. → pages 67, 68

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 127–135, 2013b. URL http://jmlr.org/proceedings/papers/v28/agrawal13.html. → pages 4, 7, 61, 68, 70

Noga Alon, Nicolo Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *arXiv preprint arXiv:1409.8428*, 2014. → page 57

R. Arratia and L. Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology*, 51(1):125–131, Jan 1989. ISSN 1522-9602. → page 153

Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo A Baeza-Yates. Online topic-aware influence maximization queries. 2014. → page 75

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002. → pages 2, 7

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *focs*, page 322. IEEE, 1995. → page 3

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. → pages 2, 3, 5, 7, 8, 59

Yixin Bao, Xiaoke Wang, Zhi Wang, Chuan Wu, and Francis C. M. Lau. Online influence maximization in non-stationary social networks. In *International Symposium on Quality of Service*, apr 2016. → page 75

Akram Baransi, Odalric-Ambrym Maillard, and Shie Mannor. Sub-sampling for multi-armed bandits. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 115–131. Springer, 2014. → page 60

Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and information systems*, 37(3):555–584, 2013. → pages 26, 35

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003. → pages 49, 138

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006. → page 33

Zahy Bnaya, Rami Puzis, Roni Stern, and Ariel Felner. Social network search as a volatile multi-armed bandit problem. *Human Journal*, 2(2):84–98, 2013. → page 40

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. → page 159

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012. URL http://arxiv.org/abs/1204.5721. → page 2

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pages 23–37. Springer, 2009. → page 77

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013. → page 71

Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012. → page 57

Alexandra Carpentier and Michal Valko. Revealing graph bandits for maximizing local influence. In *International Conference on Artificial Intelligence and Statistics*, 2016. → page 39

Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013. → pages 33, 42, 44, 45, 46, 47, 50, 52, 53, 54, 57

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011. → page 54

Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009. → page 12

Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010. → page 12

Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013a. → page 11

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013b. → page 18

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *arXiv preprint arXiv:1407.8339*, 2014. → page 77

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17, 2016a. → pages 12, 14, 39

Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016b. → pages 16, 32, 37, 39

François Chollet. keras. https://github.com/fchollet/keras, 2015. → page 71

Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011. → page 5

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc. → pages 138, 144

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 355–366, 2008. → pages 4, 5, 8, 31, 32, 52, 148

Timothy A Davis. Algorithm 849: A concise sparse cholesky factorization package. *ACM Transactions on Mathematical Software (TOMS)*, 31(4):587–591, 2005. → page 50

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. → page 57

Nan Du, Le Song, Manuel Gomez Rodriguez, and Hongyuan Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, pages 3147–3155, 2013. → page 75

Nan Du, Yingyu Liang, Maria-Florina Balcan, and Le Song. Influence Function Learning in Information Diffusion Networks. *Journal of Machine Learning Research*, 32: 2016–2024, 2014. URL
http://machinelearning.wustl.edu/mlpapers/papers/icml2014c2{_}du14. → page 12

David Easley and Jon Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010. → page 11

Dean Eckles and Maurits Kaptein. Thompson sampling with the online bootstrap. *arXiv preprint arXiv:1410.4009*, 2014. → pages 60, 61, 63, 65

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992. → page 60

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004. → page 46

Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005. → pages 33, 76

Meng Fang and Dacheng Tao. Networked bandits with disjoint linear payoffs. In *Internattional Conference on Knowledge Discovery and Data Mining*, 2014. → page 39

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23 (2):298–305, 1973. → page 49

Sarah Filippi, Olivier Cappe, Aurelien Garivier, and Csaba Szepesvari. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23*, pages 586–594, 2010a. → page 5

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010b. → pages 4, 8, 59

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. → page 62

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. → page 133

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012. → page 1

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 757–765, 2014. → pages 37, 42, 54, 56, 57

Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *International Conference on Machine Learning*, pages 1253–1262, 2017. → page 76

M Gomez Rodriguez, B Schölkopf, Langford J Pineau, et al. Influence maximization in continuous time diffusion networks. In *29th International Conference on Machine Learning (ICML 2012)*, pages 1–8. International Machine Learning Society, 2012. → pages 12, 27, 75

M. Gomez Gomez-Rodriguez and B. Schölkopf. Submodular inference of diffusion networks from multiple trees. In *ICML '12: Proceedings of the 29th International Conference on Machine Learning*, 2012. → page 11

Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Structure and dynamics of information pathways in online media. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 23–32. ACM, 2013. → page 11

Andre R Goncalves, Puja Das, Soumyadeep Chatterjee, Vidyashankar Sivakumar, Fernando J Von Zuben, and Arindam Banerjee. Multi-task sparse structure learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 451–460. ACM, 2014. → page 132

André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee. Multi-label structure learning with ising model selection. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3525–3531. AAAI Press, 2015. → page 132

Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010. → pages 12, 20, 26, 35

Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011a. → page 12

Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 211–220. IEEE, 2011b. → page 12

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016. → pages 26, 33, 75

Kai Han, Keke Huang, Xiaokui Xiao, Jing Tang, Aixin Sun, and Xueyan Tang. Efficient algorithms for adaptive influence maximization. *Proceedings of the VLDB Endowment*, 11(9):1029–1040, 2018. → page 75

Jason S. Hartford, Devon R. Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018. → page 75

Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. 1952. → pages 47, 116

Cho-Jui Hsieh, Inderjit S Dhillon, Pradeep K Ravikumar, and Mátyás A Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011. → page 133

Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173, 2013. → page 133

Kwang-Sung Jun, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. Scalable generalized linear bandits: Online computation and hashing. *arXiv preprint arXiv:1706.00136*, 2017. → pages 59, 77

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi, and Benjamin Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 3910–3919, 2017. → page 77

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003. → pages 11, 12, 27

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. → page 75

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):795–816, 2014. → page 77

Tomáš Kocák, Michal Valko, Rémi Munos, and Shipra Agrawal. Spectral thompson sampling. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. → page 57

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017. → page 77

Nathan Korda, Balázs Szörényi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1301–1309, 2016. → page 57

Samuel Kotz and Johan Ren Van Dorp. *Beyond beta: other continuous families of distributions with bounded support and applications.* World Scientific, 2004. → page 68

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012. → page 27

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a. → page 40

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pages 1450–1458, 2015b. → page 40

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015c. → page 23

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *AISTATS*, 2015d. → page 32

Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians-fast, sparse, and simple. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 573–582. IEEE, 2016. → page 50

Paul Lagrée, Olivier Cappé, Bogdan Cautis, and Silviu Maniu. Algorithms for online influencer marketing. *arXiv preprint arXiv:1702.05354*, 2017. → page 40

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985. → page 2

John Langford. The Epoch-Greedy Algorithm for Contextual Multi-armed Bandits. *Statistics*, 20:1–8, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.143.8000{&}rep=rep1{&}type=pdf. → page 7

John Langford and Tong Zhang. The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information. In J C Platt, D Koller, Y Singer, and S Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 817–824. MIT Press, Cambridge, MA, 2008. → pages 5, 6, 48, 49, 70, 135

Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 645–654, 2015. → page 39

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014. → pages 25, 36

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007. → pages 12, 34

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010. → pages 34, 53

Jurij Leskovec, Deepayan Chakrabarti, Jon Kleinberg, and Christos Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 133–145. Springer, 2005. → page 34

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. → pages 1, 4, 41, 42, 44, 48, 54

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011. → page 42

Lihong Li, Yu Lu, and Dengyong Zhou. Provable optimal algorithms for generalized linear contextual bandits. *arXiv preprint arXiv:1703.00048*, 2017. → pages 5, 59, 70

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, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 539–548, 2016. → page 57

Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 657–666. ACM, 2013. → pages 12, 27

Che-Yu Liu and Lihong Li. On the prior sensitivity of thompson sampling. In *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pages 321–336, 2016. doi:10.1007/978-3-319-46379-7_22. URL https://doi.org/10.1007/978-3-319-46379-7_22. → page 76

Vincent Yun Lou, Smriti Bhagat, Laks VS Lakshmanan, and Sharan Vaswani. Modeling non-progressive phenomena for influence propagation. In *Proceedings of the second ACM conference on Online social networks*, pages 131–138. ACM, 2014. → page 75

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. → page 56

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 136–144, 2014. URL http://jmlr.org/proceedings/papers/v32/maillard14.html. → page 57

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems*, pages 684–692, 2011. → page 57

Andreas Maurer. The rademacher complexity of linear transformation classes. In *Learning Theory*, pages 65–78. Springer, 2006. → pages 48, 49, 135, 137

Ryan McNellis, Adam N. Elmachtoub, Sechan Oh, and Marek Petrik. A practical method for solving contextual bandit problems using decision trees. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017. → pages ix, xii, 5, 60, 61, 63, 65, 69, 71, 164, 166

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001. → page 42

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978. → page 34

Seth A Myers and Jure Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 539–548. IEEE, 2012. → page 11

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978. → pages 16, 27, 28, 36

Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pages 211–222. ACM, 2012. → page 12

Michael A Newton and Adrian E Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994. → page 67

Trong T Nguyen and Hady W Lauw. Dynamic clustering of contextual multi-armed bandits. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1959–1962. ACM, 2014. → pages 42, 57

Ian Osband and Benjamin Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015. → pages 60, 61

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016. → pages 60, 77

George Papandreou and Alan L Yuille. Gaussian sampling by local perturbations. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2010. → pages 43, 48, 50

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014. → page 33

Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in Neural Information Processing Systems*, pages 2098–2106, 2015. → pages 57, 76

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018. → pages 4, 5, 60, 69

Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005. → page 132

Paat Rusmevichientong and John N Tsitsiklis. Linearly Parameterized Bandits. *Math. Oper. Res.*, 35(2):395–411, may 2010. → page 4

Avishek Saha, Piyush Rai, Suresh Venkatasubramanian, and Hal Daume. Online learning of multiple tasks and their relationships. In *International Conference on Artificial Intelligence and Statistics*, pages 643–651, 2011. → pages 132, 133

Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 67–75. Springer, 2008. → page 12

Adish Singla, Eric Horvitz, Pushmeet Kohli, Ryen White, and Andreas Krause. Information gathering in networks via active exploration. In *International Joint Conferences on Artificial Intelligence*, 2015. → page 40

Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014. → page 77

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009. → page 41

Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1587–1594. ACM, 2013. → page 1

Liang Tang, Yexi Jiang, Lei Li, Chunqiu Zeng, and Tao Li. Personalized recommendation via parameter-free contextual bandits. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 323–332. ACM, 2015a. → pages 60, 61, 63, 65, 71

Youze Tang, Xiaokui Xiao, and Shi Yanchen. Influence maximization: Near-optimal time complexity meets practical efficiency. 2014. → pages 12, 26, 36

Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International*

Conference on Management of Data, SIGMOD '15, pages 1539–1554, 2015b. ISBN 978-1-4503-2758-9. → pages 12, 16

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. → pages 1, 8

Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodriguez. Deep reinforcement learning of marked temporal point processes. *arXiv preprint arXiv:1805.09360*, 2018. → page 75

Michal Valko. *Bandits on graphs and structures.* habilitation, École normale supérieure de Cachan, 2016. → page 12

Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *31th International Conference on Machine Learning*, 2014. → pages 33, 57

Sharan Vaswani and Laks V. S. Lakshmanan. Adaptive influence maximization in social networks: Why commit when you can adapt? Technical report, 2016. → page 75

Sharan Vaswani, Laks. V. S. Lakshmanan, and Mark Schmidt. Influence maximization with bandits. Technical report, http://arxiv.org/abs/1503.00024, 2015. URL http://arxiv.org/abs/1503.00024. → pages 12, 39, 40

Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. Model-independent online learning for influence maximization. In *International Conference on Machine Learning*, pages 3530–3539, 2017a. → page iv

Sharan Vaswani, Mark Schmidt, and Laks Lakshmanan. Horde of bandits using gaussian markov random fields. In *Artificial Intelligence and Statistics*, pages 690–699, 2017b. → pages v, 33, 117

Sharan Vaswani, Branislav Kveton, Zheng Wen, Anup Rao, Mark Schmidt, and Yasin Abbasi-Yadkori. New insights into bootstrapping for bandits. *arXiv preprint arXiv:1805.09793*, 2018. → page v

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007. → page 37

Qinshi Wang and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Neural Information Processing Systems*, mar 2017. URL http://arxiv.org/abs/1703.01610. → page 39

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a. → page 23

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1113–1122, 2015b. → pages 32, 52, 148

Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *Advances in Neural Information Processing Systems*, pages 3022–3032, 2017. → page iv

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *International Conference on Machine Learning*, pages 1254–1262, 2016. → page 77

Amulya Yadav, Bryan Wilder, Eric Rice, Robin Petering, Jaih Craddock, Amanda Yoshioka-Maxwell, Mary Hemler, Laura Onasch-Vera, Milind Tambe, and Darlene Woo. Influence maximization in the field: The arduous journey from emerging to deployed application. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 150–158. International Foundation for Autonomous Agents and Multiagent Systems, 2017. → page 12

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 392–401, 2016. URL http://jmlr.org/proceedings/papers/v48/zhangb16.html. → page 59

Xiaojin Zhu. Semi-supervised learning with graphs. 2005. → page 76

Appendix A

Supplementary for Chapter 2

A.1 Proof of Theorem 1

In the appendix, we prove a slightly stronger version of Theorem 1, which also uses another complexity metric E_* defined as follows: Assume that the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ includes m disconnected subgraphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1), \mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2), \dots, \mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$, which are in the descending order based on the number of nodes $|\mathcal{E}_i|$'s. We define E_* as the number of edges in the first $\min\{m, K\}$ subgraphs:

$$E_* = \sum_{i=1}^{\min\{m,K\}} |\mathcal{E}_i|. \tag{A.1}$$

Note that by definition, $E_* \leq m$. Based on E_*, we have the following slightly stronger version of Theorem 1.

Theorem 8. *Assume that (1) $p(e) = x_e^{\mathsf{T}} \theta^*$ for all $e \in \mathcal{E}$ and (2)* `ORACLE` *is an (α, γ)-approximation algorithm. Let D be a known upper bound on $\|\theta^*\|_2$. If we apply* `ICLinUCB` *with $\sigma = 1$ and*

$$c \geq \sqrt{d \log\left(1 + \frac{TE_*}{d}\right) + 2\log\left(T(n+1-K)\right)} + D, \tag{A.2}$$

then we have

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma}\sqrt{dTE_* \log_2\left(1+\frac{TE_*}{d}\right)}+1 = \widetilde{\mathcal{O}}\left(dC_*\sqrt{E_*T}/(\alpha\gamma)\right). \tag{A.3}$$

Moreover, if the feature matrix is of the form $X = I \in \Re^{m \times m}$ (i.e., the tabular case), we have

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma}\sqrt{Tm \log_2\left(1+T\right)}+1 = \widetilde{\mathcal{O}}\left(mC_*\sqrt{T}/(\alpha\gamma)\right). \tag{A.4}$$

Since $E_* \leq m$, Theorem 8 implies Theorem 1. We prove Theorem 8 in the remainder of this section.

We now define some notation to simplify the exposition throughout this section.

Definition 1. *For any source node set $\mathcal{S} \subseteq \mathcal{V}$, any probability weight function $w : \mathcal{E} \to [0, 1]$, and any node $v \in \mathcal{V}$, we define $f(\mathcal{S}, w, v)$ as the probability that node v is influenced if the source node set is $\mathcal{S}$ and the probability weight function is w.*

Notice that by definition, $f(\mathcal{S}, w) = \sum_{v\in\mathcal{V}} f(\mathcal{S}, w, v)$ always holds. Moreover, if $v \in \mathcal{S}$, then $f(\mathcal{S}, w, v) = 1$ for any w by the definition of the influence model.

Definition 2. *For any round t and any directed edge $e \in \mathcal{E}$, we define event*

$$O_t(e) = \{\textit{edge } e \textit{ is observed at round } t\}.$$

Note that by definition, an directed edge e is observed if and only if its start node is influenced and observed does not necessarily mean that the edge is *active*.

A.1.1 Proof of Theorem 8

Proof. Let $\mathcal{H}_t$ be the history (σ-algebra) of past observations and actions by the end of round t. By the definition of $R_t^{\alpha\gamma}$, we have

$$\mathbb{E}\left[R_t^{\alpha\gamma}|\mathcal{H}_{t-1}\right] = f(\mathcal{S}^*, p) - \frac{1}{\alpha\gamma}\mathbb{E}\left[f(\mathcal{S}_t, p)\right|\mathcal{H}_{t-1}], \tag{A.5}$$

where the expectation is over the possible randomness of $\mathcal{S}_t$, since `ORACLE` might be a randomized algorithm. Notice that the randomness coming from the edge activation is already taken care of in the definition of f. For any $t \leq T$, we define event ξ_{t-1} as

$$\xi_{t-1} = \left\{ |x_e^\mathsf{T}(\bar{\theta}_{\tau-1} - \theta^*)| \leq c\sqrt{x_e^\mathsf{T} \mathbf{M}_{\tau-1}^{-1} x_e}, \, \forall e \in \mathcal{E}, \, \forall \tau \leq t \right\}, \tag{A.6}$$

and $\bar{\xi}_{t-1}$ as the complement of ξ_{t-1}. Notice that ξ_{t-1} is $\mathcal{H}_{t-1}$-measurable. Hence we have

$$\mathbb{E}[R_t^{\alpha\gamma}] \leq \mathbb{P}\left(\xi_{t-1}\right) \mathbb{E}\left[f(\mathcal{S}^*, p) - f(\mathcal{S}_t, p)/(\alpha\gamma) | \xi_{t-1}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n - K].$$

Notice that under event ξ_{t-1}, $p(e) \leq U_t(e)$, $\forall e \in \mathcal{E}$, for all $t \leq T$, thus we have

$$f(\mathcal{S}^*, p) \leq f(\mathcal{S}^*, U_t) \leq \max_{\mathcal{S}\colon |\mathcal{S}| = K} f(\mathcal{S}, U_t) \leq \frac{1}{\alpha\gamma} \mathbb{E}\left[f(\mathcal{S}_t, U_t) \middle| \mathcal{H}_{t-1}\right],$$

where the first inequality follows from the monotonicity of f in the probability weight, and the last inequality follows from the fact that `ORACLE` is an (α, γ)-approximation algorithm. Thus, we have

$$\mathbb{E}[R_t^{\alpha\gamma}] \leq \frac{\mathbb{P}\left(\xi_{t-1}\right)}{\alpha\gamma} \mathbb{E}\left[f(\mathcal{S}_t, U_t) - f(\mathcal{S}_t, p) | \xi_{t-1}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n - K]. \tag{A.7}$$

Notice that based on Definition 1, we have

$$f(\mathcal{S}_t, U_t) - f(\mathcal{S}_t, p) = \sum_{v \in \mathcal{V} \setminus \mathcal{S}_t} \left[f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v)\right].$$

Recall that for a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a given source node set $\mathcal{S} \subseteq \mathcal{V}$, we say an edge $e \in \mathcal{E}$ and a node $v \in \mathcal{V} \setminus \mathcal{S}$ are *relevant* if there exists a path p from a source node $s \in \mathcal{S}$ to v such that (1) $e \in p$ and (2) p does not contain another source node other than s. We use $\mathcal{E}_{\mathcal{S},v} \subseteq \mathcal{E}$ to denote the set of edges relevant to node v under the source node set $\mathcal{S}$, and use $\mathcal{V}_{\mathcal{S},v} \subseteq \mathcal{V}$ to denote the set of nodes connected to at least one edge in $\mathcal{E}_{\mathcal{S},v}$. Notice that $\mathcal{G}_{\mathcal{S},v} \stackrel{\Delta}{=} (\mathcal{V}_{\mathcal{S},v}, \mathcal{E}_{\mathcal{S},v})$ is a subgraph of $\mathcal{G}$, and we refer to it as the **relevant subgraph** of node v under the source node set $\mathcal{S}$.

Based on the notion of relevant subgraph, we have the following theorem, which bounds $f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v)$ by edge-level gaps $U_t(e) - p(e)$ on the observed edges in the relevant subgraph $\mathcal{G}_{\mathcal{S}_t,v}$ for node v;

Theorem 9. *For any t, any history $\mathcal{H}_{t-1}$ and $\mathcal{S}_t$ such that ξ_{t-1} holds, and any $v \in \mathcal{V} \setminus \mathcal{S}_t$,* *we have*

$$f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v) \leq \sum_{e \in \mathcal{E}_{\mathcal{S}_t, v}} \mathbb{E}\left[\mathbf{1}\left\{O_t(e)\right\}\left[U_t(e) - p(e)\right] | \mathcal{H}_{t-1}, \mathcal{S}_t\right],$$

where $\mathcal{E}_{\mathcal{S}_t,v}$ is the edge set of the relevant subgraph $\mathcal{G}_{\mathcal{S}_t,v}$.

Please refer to Section A.1.2 for the proof of Theorem 9. Notice that under favorable event ξ_{t-1}, we have $U_t(e) - p(e) \leq 2c\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}$ for all $e \in \mathcal{E}$. Therefore, we have

$$\begin{aligned}
\mathbb{E}[R_t^{\alpha\gamma}] &\leq \frac{2c}{\alpha\gamma}\mathbb{P}\left(\xi_{t-1}\right)\mathbb{E}\left[\sum_{v \in \mathcal{V} \setminus \mathcal{S}_t} \sum_{e \in \mathcal{E}_{\mathcal{S}_t,v}} \mathbf{1}\{O_t(e)\}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \middle| \xi_{t-1}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n-K] \\
&\leq \frac{2c}{\alpha\gamma}\mathbb{E}\left[\sum_{v \in \mathcal{V} \setminus \mathcal{S}_t} \sum_{e \in \mathcal{E}_{\mathcal{S}_t,v}} \mathbf{1}\{O_t(e)\}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n-K] \\
&= \frac{2c}{\alpha\gamma}\mathbb{E}\left[\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \sum_{v \in \mathcal{V} \setminus \mathcal{S}_t} \mathbf{1}\left\{e \in \mathcal{E}_{\mathcal{S}_t,v}\right\}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n-K] \\
&= \frac{2c}{\alpha\gamma}\mathbb{E}\left[\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}\right] + \mathbb{P}\left(\bar{\xi}_{t-1}\right)[n-K], \qquad \text{(A.8)}
\end{aligned}$$

where $N_{\mathcal{S}_t,e} = \sum_{v \in \mathcal{V} \setminus \mathcal{S}} \mathbf{1}\left\{e \in \mathcal{E}_{\mathcal{S}_t,v}\right\}$ is defined in Equation 2.2. Thus we have

$$R^{\alpha\gamma}(T) \leq \frac{2c}{\alpha\gamma}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}\right] + [n-K]\sum_{t=1}^{T}\mathbb{P}\left(\bar{\xi}_{t-1}\right). \qquad \text{(A.9)}$$

In the following lemma, we give a worst-case bound on $\sum_{t=1}^{T}\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}$.

Lemma 2. *For any round $t = 1, 2, \ldots, T$, we have*

$$\sum_{t=1}^{T}\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \leq \sqrt{\left(\sum_{t=1}^{T}\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2\right)\frac{dE_* \log\left(1 + \frac{TE_*}{d\sigma^2}\right)}{\log\left(1 + \frac{1}{\sigma^2}\right)}}.$$

Moreover, if $X = I \in \Re^{m \times m}$, then we have

$$\sum_{t=1}^{T} \sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e} \sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \leq \sqrt{\left(\sum_{t=1}^{T} \sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2 \right) \frac{m \log\left(1 + \frac{T}{\sigma^2}\right)}{\log\left(1 + \frac{1}{\sigma^2}\right)}}.$$

Please refer to Section A.1.3 for the proof of Lemma 2. Finally, notice that for any t,

$$\mathbb{E}\left[\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2 \middle| \mathcal{S}_t \right] = \sum_{e \in \mathcal{E}} N_{\mathcal{S}_t,e}^2 \mathbb{E}\left[\mathbf{1}\{O_t(e)\} | \mathcal{S}_t\right] = \sum_{e \in \mathcal{E}} N_{\mathcal{S}_t,e}^2 P_{\mathcal{S}_t,e} \leq C_*^2,$$

thus taking the expectation over the possibly randomized oracle and Jensen's inequality, we get

$$\mathbb{E}\left[\sqrt{\sum_{t=1}^{T} \sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2} \right] \leq \sqrt{\sum_{t=1}^{T} \mathbb{E}\left[\sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2 \right]} \leq \sqrt{\sum_{t=1}^{T} C_*^2} = C_* \sqrt{T}. \tag{A.10}$$

Combining the above with Lemma 2 and (A.9), we obtain

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma} \sqrt{\frac{dTE_* \log\left(1 + \frac{TE_*}{d\sigma^2}\right)}{\log\left(1 + \frac{1}{\sigma^2}\right)}} + [n - K] \sum_{t=1}^{T} \mathbb{P}\left(\overline{\xi}_{t-1}\right). \tag{A.11}$$

For the special case when $X = I$, we have

$$R^{\alpha\gamma}(T) \leq \frac{2cC_*}{\alpha\gamma} \sqrt{\frac{Tm \log\left(1 + \frac{T}{\sigma^2}\right)}{\log\left(1 + \frac{1}{\sigma^2}\right)}} + [n - K] \sum_{t=1}^{T} \mathbb{P}\left(\overline{\xi}_{t-1}\right). \tag{A.12}$$

Finally, we need to bound the failure probability of upper confidence bound being wrong $\sum_{t=1}^{T} \mathbb{P}\left(\overline{\xi}_{t-1}\right)$. We prove the following bound on $\mathbb{P}\left(\overline{\xi}_{t-1}\right)$:

Lemma 3. *For any $t = 1, 2, \ldots, T$, any $\sigma > 0$, any $\delta \in (0, 1)$, and any*

$$c \geq \frac{1}{\sigma} \sqrt{d \log\left(1 + \frac{TE_*}{d\sigma^2}\right) + 2 \log\left(\frac{1}{\delta}\right)} + \|\theta^*\|_2,$$

we have $\mathbb{P}\left(\overline{\xi}_{t-1}\right) \leq \delta$.

Please refer to Section A.1.4 for the proof of Lemma 3. From Lemma 3, for a known upper bound D on $\|\theta^*\|_2$, if we choose $\sigma = 1$ and $c \geq \sqrt{d \log\left(1 + \frac{TE_*}{d}\right) + 2\log\left(T(n+1-K)\right)} + D$, which corresponds to $\delta = \frac{1}{T(n+1-K)}$ in Lemma 3, then we have

$$[n-K] \sum_{t=1}^{T} \mathbb{P}\left(\overline{\xi}_{t-1}\right) < 1.$$

This concludes the proof of Theorem 8. □

A.1.2 Proof of Theorem 9

Recall that we use $\mathcal{G}_{\mathcal{S}_t,v} = (\mathcal{V}_{\mathcal{S}_t,v}, \mathcal{E}_{\mathcal{S}_t,v})$ to denote the relevant subgraph of node v under the source node set $\mathcal{S}_t$. Since Theorem 9 focuses on the influence from $\mathcal{S}_t$ to v, and by definition all the paths from $\mathcal{S}_t$ to v are in $\mathcal{G}_{\mathcal{S}_t,v}$, thus, it is sufficient to restrict to $\mathcal{G}_{\mathcal{S}_t,v}$ and ignore other parts of $\mathcal{G}$ in this analysis.

We start by defining some useful notations.

Influence Probability with Removed Nodes: Recall that for any weight function $w : \mathcal{E} \to [0,1]$, any source node set $\mathcal{S} \subset \mathcal{V}$ and any target node $v \in \mathcal{V}$, $f(\mathcal{S}, w, v)$ is the probability that $\mathcal{S}$ will influence v under weight w (see Definition 1). We now define a similar notation for the **influence probability with removed nodes**. Specifically, for any disjoint node set $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}_{\mathcal{S}_t,v} \subseteq \mathcal{V}$, we define $h(\mathcal{V}_1, \mathcal{V}_2, w)$ as follows:

- First, we remove nodes $\mathcal{V}_2$, as well as all edges connected to/from $\mathcal{V}_2$, from $\mathcal{G}_{\mathcal{S}_t,v}$, and obtain a new graph $\mathcal{G}'$.
- $h(\mathcal{V}_1, \mathcal{V}_2, w)$ is the probability that $\mathcal{V}_1$ will influence the target node v in graph $\mathcal{G}'$ under the weight (activation probability) $w(e)$ for all $e \in \mathcal{G}'$.

Obviously, a mathematically equivalent way to define $h(\mathcal{V}_1, \mathcal{V}_2, w)$ is to define it as the probability that $\mathcal{V}_1$ will influence v in $\mathcal{G}_{\mathcal{S}_t,v}$ under a new weight $\widetilde{w}$, defined as

$$\widetilde{w}(e) = \begin{cases} 0 & \text{if } e \text{ is from or to a node in } \mathcal{V}_2 \\ w(e) & \text{otherwise} \end{cases}$$

Note that by definition, $f(\mathcal{S}_t, w, v) = h(\mathcal{S}_t, \emptyset, w)$. Also note that $h(\mathcal{V}_1, \mathcal{V}_2, w)$ implicitly depends on v, but we omit v in this notation to simplify the exposition.

Edge Set $\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2)$: For any two disjoint node sets $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}_{\mathcal{S}_t,v}$, we define the edge set $\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2)$ as

$$\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2) = \{e = (u_1, u_2) : e \in \mathcal{E}_{\mathcal{S}_t,v},\, u_1 \in \mathcal{V}_1, \text{ and } u_2 \notin \mathcal{V}_2\}.$$

That is, $\mathcal{E}(\mathcal{V}_1, \mathcal{V}_2)$ is the set of edges in $\mathcal{G}_{\mathcal{S}_t,v}$ from $\mathcal{V}_1$ to $\mathcal{V}_{\mathcal{S}_t,v} \setminus \mathcal{V}_2$.

Diffusion Process: Note that under any edge activation realization $\mathbf{w}(e)$, $e \in \mathcal{E}_{\mathcal{S}_t,v}$, on the relevant subgraph $\mathcal{G}_{\mathcal{S}_t,v}$, we define a finite-length sequence of disjoint node sets $\mathcal{S}^0, \mathcal{S}^1, \ldots, \mathcal{S}^{\widetilde{\tau}}$ as

$$\begin{aligned} \mathcal{S}^0 &\stackrel{\Delta}{=} \mathcal{S}_t \\ \mathcal{S}^{\tau+1} &\stackrel{\Delta}{=} \left\{ u_2 \in \mathcal{V}_{\mathcal{S}_t,v} : u_2 \notin \cup_{\tau'=0}^{\tau} \mathcal{S}^{\tau'} \text{ and } \exists e = (u_1, u_2) \in \mathcal{E}_{\mathcal{S}_t,v} \text{ s.t. } u_1 \in \mathcal{S}^{\tau} \text{ and } \mathbf{w}(e) = 1 \right\}, \end{aligned} \tag{A.13}$$

$\forall \tau = 0, \ldots, \widetilde{\tau} - 1$. That is, under the realization $\mathbf{w}(e)$, $e \in \mathcal{E}_{\mathcal{S}_t,v}$, $\mathcal{S}^{\tau+1}$ is the set of nodes directly activated by $\mathcal{S}^{\tau}$. Specifically, any node $u_2 \in \mathcal{S}^{\tau+1}$ satisfies $u_2 \notin \bigcup_{\tau'=0}^{\tau} \mathcal{S}^{\tau'}$ (i.e. it was not activated before), and there exists an activated edge e from $\mathcal{S}^{\tau}$ to u_2 (i.e. it is activated by some node in $\mathcal{S}^{\tau}$). We define $\mathcal{S}^{\widetilde{\tau}}$ as the first node set in the sequence s.t. either $\mathcal{S}^{\widetilde{\tau}} = \emptyset$ or $v \in \mathcal{S}^{\widetilde{\tau}}$, and assume this sequence terminates at $\mathcal{S}^{\widetilde{\tau}}$. Note that by definition, $\widetilde{\tau} \leq |\mathcal{V}_{\mathcal{S}_t,v}|$ always holds. We refer to each $\tau = 0, 1, \ldots, \widetilde{\tau}$ as a **diffusion step** in this section.

To simplify the exposition, we also define $S^{0:\tau} \stackrel{\Delta}{=} \bigcup_{\tau'=0}^{\tau} S^{\tau'}$ for all $\tau \geq 0$ and $S^{0:-1} \stackrel{\Delta}{=} \emptyset$. Since $\mathbf{w}$ is random, $(\mathcal{S}^{\tau})_{\tau=0}^{\widetilde{\tau}}$ is a stochastic process, which we refer to as the **diffusion process**. Note that $\widetilde{\tau}$ is also random; in particular, it is a stopping time.

Based on the shorthand notations defined above, we have the following lemma for the diffusion process $(\mathcal{S}^{\tau})_{\tau=0}^{\widetilde{\tau}}$ under any weight function w:

Lemma 4. *For any weight function $w : \mathcal{E} \to [0, 1]$, any step $\tau = 0, 1, \ldots, \widetilde{\tau}$, any $\mathcal{S}_{\tau}$ and*

$\mathcal{S}^{0:\tau-1}$*, we have*

$$h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)=\begin{cases}1 & \textit{if } v\in\mathcal{S}^{\tau}\\ 0 & \textit{if } \mathcal{S}^{\tau}=\emptyset\\ \mathbb{E}\left[h\left(\mathcal{S}^{\tau+1},\mathcal{S}^{0:\tau},w\right)\big|(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1})\right] & \textit{otherwise}\end{cases},$$

where the expectation is over $\mathcal{S}^{\tau+1}$ *under weight* w*. Note that the tuple* $(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1})$ *in the conditional expectation means that* $\mathcal{S}^{\tau}$ *is the source node set and nodes in* $\mathcal{S}^{0:\tau-1}$ *have been removed.*

Proof. Notice that by definition, $h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)=1$ if $v\in\mathcal{S}^{\tau}$ and $h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)=0$ if $\mathcal{S}^{\tau}=\emptyset$. Also note that in these two cases, $\widetilde{\tau}=\tau$.

Otherwise, we prove that $h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)=\mathbb{E}\left[h\left(\mathcal{S}^{\tau+1},\mathcal{S}^{0:\tau},w\right)\big|(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1})\right]$. Recall that by definition, $h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)$ is the probability that v will be influenced conditioning on

$$\text{source node set } \mathcal{S}^{\tau} \text{ and removed node set } \mathcal{S}^{0:\tau-1}, \tag{A.14}$$

that is

$$h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w\right)=\mathbb{E}\left[\mathbf{1}\left(v \text{ is influenced}\right)\big|(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1})\right] \tag{A.15}$$

Let $\mathbf{w}(e)$, $\forall e\in\mathcal{E}(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau})$ be any possible realization. Now we analyze the probability that v will be influenced conditioning on

$$\text{source node set } \mathcal{S}^{\tau}, \text{ removed node set } \mathcal{S}^{0:\tau-1}, \text{ and } \mathbf{w}(e) \text{ for all } e\in\mathcal{E}(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau}). \tag{A.16}$$

Specifically, conditioning on Equation A.16, we can define a new weight function w' as

$$w'(e)=\begin{cases}\mathbf{w}(e) & \text{if } e\in\mathcal{E}(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau})\\ w(e) & \text{otherwise}\end{cases} \tag{A.17}$$

then $h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w'\right)$ is the probability that v will be influenced conditioning on Equation A.16. That is,

$$h\left(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1},w'\right)=\mathbb{E}\left[\mathbf{1}\left(v \text{ is influenced}\right)\big|(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau-1}),\mathbf{w}(e)\,\forall e\in\mathcal{E}(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau})\right], \tag{A.18}$$

for any possible realization of $\mathbf{w}(e)$, $\forall e\in\mathcal{E}(\mathcal{S}^{\tau},\mathcal{S}^{0:\tau})$. Notice that on the lefthand of Equation A.18, w' encodes the conditioning on $\mathbf{w}(e)$ for all $e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$ (see Equation A.17).

From here to Equation A.20, we focus on an arbitrary but fixed realization of $\mathbf{w}(e)$, $\forall e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$ (or equivalently, an arbitrary but fixed w'). Based on the definition of $\mathcal{S}^{\tau+1}$, conditioning on Equation A.16, $\mathcal{S}^{\tau+1}$ is deterministic and all nodes in $\mathcal{S}^{\tau+1}$ can also be treated as source nodes. Thus, we have

$$h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w'\right) = h\left(\mathcal{S}^\tau \cup \mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau-1}, w'\right),$$

conditioning on Equation A.16.

On the other hand, conditioning on Equation A.16, we can treat any edge $e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$ with $\mathbf{w}(e) = 0$ as having been removed. Since nodes in $\mathcal{S}^{0:\tau-1}$ have also been removed, and $v \notin \mathcal{S}^\tau$, then if there is a path from $\mathcal{S}^\tau$ to v, then it must go through $\mathcal{S}^{\tau+1}$, and the last node on the path in $\mathcal{S}^{\tau+1}$ must be after the last node on the path in $\mathcal{S}^\tau$ (note that the path might come back to $\mathcal{S}^\tau$ for several times). Hence, conditioning on Equation A.16, if nodes in $\mathcal{S}^{\tau+1}$ are also treated as source nodes, then $\mathcal{S}^\tau$ is irrelevant for influence on v and can be removed. So we have

$$h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w'\right) = h\left(\mathcal{S}^\tau \cup \mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau-1}, w'\right) = h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right). \tag{A.19}$$

Note that in the last equation we change the weight function back to w since edges in $\mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$ have been removed. Thus, conditioning on Equation A.16, we have

$$\begin{aligned} h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right) &= h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w'\right) \\ &= \mathbb{E}\left[\mathbf{1}\left(v \text{ is influenced}\right) \middle| (\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}), \mathbf{w}(e)\, \forall e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})\right]. \end{aligned} \tag{A.20}$$

Notice again that Equation A.20 holds for any possible realization of $\mathbf{w}(e)$, $\forall e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$.

Finally, we have

$$\begin{aligned} h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w\right) &\overset{(a)}{=} \mathbb{E}\left[\mathbf{1}\left(v \text{ is influenced}\right) \middle| (\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})\right] \\ &\overset{(b)}{=} \mathbb{E}\left[\mathbb{E}\left[\mathbf{1}\left(v \text{ is influenced}\right) \middle| (\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}), \mathbf{w}(e)\, \forall e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})\right] \middle| (\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})\right] \\ &\overset{(c)}{=} \mathbb{E}\left[h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right) \middle| (\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})\right], \end{aligned} \tag{A.21}$$

where (a) follows from Equation A.15, (b) follows from the tower rule, and (c) follows from

Equation A.20. This concludes the proof. □

Consider two weight functions $U, w : \mathcal{E} \to [0,1]$ s.t. $U(e) \geq w(e)$ for all $e \in \mathcal{E}$. The following lemma bounds the difference $h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right)$ in a recursive way.

Lemma 5. *For any two weight functions $w, U : \mathcal{E} \to [0,1]$ s.t. $U(e) \geq w(e)$ for all $e \in \mathcal{E}$, any step $\tau = 0, 1, \ldots, \widetilde{\tau}$, any $\mathcal{S}_{\tau}$ and $\mathcal{S}^{0:\tau-1}$, we have*

$$h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right) = 0$$

if $v \in \mathcal{S}^{\tau}$ or $\mathcal{S}^{\tau} = \emptyset$; and otherwise

$$\begin{aligned} h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right) \leq & \sum_{e \in \mathcal{E}\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau}\right)} [U(e) - w(e)] \\ & + \mathbb{E}\left[h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right) \middle| (\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1})\right], \end{aligned}$$

where the expectation is over $\mathcal{S}^{\tau+1}$ under weight w. Recall that the tuple $(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1})$ in the conditional expectation means that $\mathcal{S}^{\tau}$ is the source node set and nodes in $\mathcal{S}^{0:\tau-1}$ have been removed.

Proof. First, note that if $v \in \mathcal{S}^{\tau}$ or $\mathcal{S}^{\tau} = \emptyset$, then

$$h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right) = 0$$

follows directly from Lemma 4. Otherwise, to simplify the exposition, we overload the notation and use $w(\mathcal{S}^{\tau+1})$ to denote the conditional probability of $\mathcal{S}^{\tau+1}$ conditioning on $(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1})$ under the weight function w, and similarly for $U(\mathcal{S}^{\tau+1})$. That is

$$\begin{aligned} w(\mathcal{S}^{\tau+1}) & \stackrel{\Delta}{=} \text{Prob}\left[\mathcal{S}^{\tau+1} \middle| (\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}); w\right] \\ U(\mathcal{S}^{\tau+1}) & \stackrel{\Delta}{=} \text{Prob}\left[\mathcal{S}^{\tau+1} \middle| (\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}); U\right], \end{aligned} \tag{A.22}$$

where the tuple $(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1})$ in the conditional probability means that $\mathcal{S}^{\tau}$ is the source node set and nodes in $\mathcal{S}^{0:\tau-1}$ have been removed, and w and U after the semicolon indicate the weight function.

Then from Lemma 4, we have

$$h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) = \sum_{\mathcal{S}^{\tau+1}} U(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right)$$
$$h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right) = \sum_{\mathcal{S}^{\tau+1}} w(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right)$$

where the sum is over all possible realization of $\mathcal{S}^{\tau+1}$.

Hence we have

$$\begin{aligned}
& h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1}, w\right) \\
= & \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - w(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right)\right] \\
= & \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - w(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right)\right] \\
+ & \sum_{\mathcal{S}^{\tau+1}} \left[w(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - w(\mathcal{S}^{\tau+1}) h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right)\right] \\
= & \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \\
+ & \sum_{\mathcal{S}^{\tau+1}} w(\mathcal{S}^{\tau+1}) \left[h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right)\right],
\end{aligned} \tag{A.23}$$

where the sum in the above equations is also over all the possible realizations of $\mathcal{S}^{\tau+1}$. Notice that by definition, we have

$$\begin{aligned}
& \mathbb{E}\left[h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right) \middle| (\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau-1})\right] = \\
& \qquad \sum_{\mathcal{S}^{\tau+1}} w(\mathcal{S}^{\tau+1}) \left[h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) - h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, w\right)\right],
\end{aligned} \tag{A.24}$$

where the expectation in the lefthand side is over $\mathcal{S}^{\tau+1}$ under weight w, or equivalently, over $\mathbf{w}(e)$ for all $e \in \mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})$ under weight w. Thus, to prove Lemma 5, it is sufficient to prove that

$$\sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \leq \sum_{e \in \mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})} \left[U(e) - w(e)\right]. \tag{A.25}$$

Notice that

$$\begin{aligned}
&\sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \\
\overset{(a)}{\leq} &\sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) \geq w(\mathcal{S}^{\tau+1})\right] \\
\overset{(b)}{\leq} &\sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) \geq w(\mathcal{S}^{\tau+1})\right] \\
\overset{(c)}{=} &\frac{1}{2} \sum_{\mathcal{S}^{\tau+1}} \left|U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right|, \qquad \text{(A.26)}
\end{aligned}$$

where (a) holds since

$$\begin{aligned}
&\sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) = \\
&\qquad \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) \geq w(\mathcal{S}^{\tau+1})\right] \\
&\qquad\qquad + \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) < w(\mathcal{S}^{\tau+1})\right],
\end{aligned}$$

and the second term on the righthand side is non-positive. And (b) holds since $0 \leq h\left(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau}, U\right) \leq 1$ by definition. To prove (c), we define shorthand notations

$$\begin{aligned}
A^+ &= \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) \geq w(\mathcal{S}^{\tau+1})\right] \\
A^- &= \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] \mathbf{1}\left[U(\mathcal{S}^{\tau+1}) < w(\mathcal{S}^{\tau+1})\right]
\end{aligned}$$

Then we have

$$A^+ + A^- = \sum_{\mathcal{S}^{\tau+1}} \left[U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right] = 0,$$

since by definition $\sum_{\mathcal{S}^{\tau+1}} U(\mathcal{S}^{\tau+1}) = \sum_{\mathcal{S}^{\tau+1}} w(\mathcal{S}^{\tau+1}) = 1$. Moreover, we also have

$$A^+ - A^- = \sum_{\mathcal{S}^{\tau+1}} \left|U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right|.$$

And hence $A^+ = \frac{1}{2}\sum_{\mathcal{S}^{\tau+1}} \left|U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right|$. Thus, to prove Lemma 5, it is sufficient to prove

$$\frac{1}{2}\sum_{\mathcal{S}^{\tau+1}} \left|U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right| \leq \sum_{e \in \mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})} \left[U(e) - w(e)\right]. \tag{A.27}$$

Let $\widetilde{\mathbf{w}} \in \{0,1\}^{|\mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})|}$ be an arbitrary edge activation realization for edges in $\mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})$. Also with a little bit abuse of notation, we use $w(\widetilde{\mathbf{w}})$ to denote the probability of $\widetilde{\mathbf{w}}$ under weight w. Notice that

$$w(\widetilde{\mathbf{w}}) = \prod_{e \in \mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})} w(e)^{\widetilde{\mathbf{w}}(e)} \left[1 - w(e)\right]^{1 - \widetilde{\mathbf{w}}(e)},$$

and $U(\widetilde{\mathbf{w}})$ is defined similarly. Recall that by definition $\mathcal{S}^{\tau+1}$ is a deterministic function of source node set $\mathcal{S}^{\tau}$, removed nodes $\mathcal{S}^{0:\tau-1}$, and $\widetilde{\mathbf{w}}$. Hence, for any possible realized $\mathcal{S}^{\tau+1}$, let $\mathbf{W}(\mathcal{S}^{\tau+1})$ denote the set of $\widetilde{\mathbf{w}}$'s that lead to this $\mathcal{S}^{\tau+1}$, then we have

$$U(\mathcal{S}^{\tau+1}) = \sum_{\widetilde{\mathbf{w}} \in \mathbf{W}(\mathcal{S}^{\tau+1})} U(\widetilde{\mathbf{w}}) \quad \text{and} \quad w(\mathcal{S}^{\tau+1}) = \sum_{\widetilde{\mathbf{w}} \in \mathbf{W}(\mathcal{S}^{\tau+1})} w(\widetilde{\mathbf{w}})$$

Thus, we have

$$\begin{aligned}
\frac{1}{2}\sum_{\mathcal{S}^{\tau+1}} \left|U(\mathcal{S}^{\tau+1}) - w(\mathcal{S}^{\tau+1})\right| =& \frac{1}{2}\sum_{\mathcal{S}^{\tau+1}} \left| \sum_{\widetilde{\mathbf{w}} \in \mathbf{W}(\mathcal{S}^{\tau+1})} \left[U(\widetilde{\mathbf{w}}) - w(\widetilde{\mathbf{w}})\right] \right| \\
\leq& \frac{1}{2}\sum_{\mathcal{S}^{\tau+1}} \sum_{\widetilde{\mathbf{w}} \in \mathbf{W}(\mathcal{S}^{\tau+1})} \left|U(\widetilde{\mathbf{w}}) - w(\widetilde{\mathbf{w}})\right| \\
=& \frac{1}{2}\sum_{\widetilde{\mathbf{w}}} \left|U(\widetilde{\mathbf{w}}) - w(\widetilde{\mathbf{w}})\right| \qquad \text{(A.28)}
\end{aligned}$$

Finally, we prove that

$$\frac{1}{2}\sum_{\widetilde{\mathbf{w}}} \left|U(\widetilde{\mathbf{w}}) - w(\widetilde{\mathbf{w}})\right| \leq \sum_{e \in \mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})} \left[U(e) - w(e)\right] \tag{A.29}$$

by mathematical induction. Without loss of generality, we order the edges in $\mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})$ as $1, 2, \ldots, |\mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})|$. For any $k = 1, \ldots, |\mathcal{E}(\mathcal{S}^{\tau}, \mathcal{S}^{0:\tau})|$, we use $\widetilde{\mathbf{w}}_k \in \{0,1\}^k$ to denote an arbitrary edge activation realization for edges $1, \ldots, k$. Then, we prove

$$\frac{1}{2} \sum_{\widetilde{\mathbf{w}}_k} |U(\widetilde{\mathbf{w}}_k) - w(\widetilde{\mathbf{w}}_k)| \leq \sum_{e=1}^{k} [U(e) - w(e)] \tag{A.30}$$

for all $k = 1, \ldots, |\mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})|$ by mathematical induction. Notice that when $k = 1$, we have

$$\frac{1}{2} \sum_{\widetilde{\mathbf{w}}_1} |U(\widetilde{\mathbf{w}}_1) - w(\widetilde{\mathbf{w}}_1)| = \frac{1}{2} [|U(1) - w(1)| + |(1 - U(1)) - (1 - w(1))|] = U(1) - w(1).$$

Now assume that the induction hypothesis holds for k, we prove that it also holds for $k+1$. Note that

$$\begin{aligned}
\frac{1}{2} \sum_{\widetilde{\mathbf{w}}_{k+1}} |U(\widetilde{\mathbf{w}}_{k+1}) - w(\widetilde{\mathbf{w}}_{k+1})| =& \frac{1}{2} \sum_{\widetilde{\mathbf{w}}_k} [|U(\widetilde{\mathbf{w}}_k)U(k+1) - w(\widetilde{\mathbf{w}}_k)w(k+1)| \\
&+ |U(\widetilde{\mathbf{w}}_k)(1 - U(k+1)) - w(\widetilde{\mathbf{w}}_k)(1 - w(k+1))|] \\
\overset{(a)}{\leq}& \frac{1}{2} \sum_{\widetilde{\mathbf{w}}_k} [|U(\widetilde{\mathbf{w}}_k)U(k+1) - w(\widetilde{\mathbf{w}}_k)U(k+1)| \\
&+ |w(\widetilde{\mathbf{w}}_k)U(k+1) - w(\widetilde{\mathbf{w}}_k)w(k+1)| \\
&+ |U(\widetilde{\mathbf{w}}_k)(1 - U(k+1)) - w(\widetilde{\mathbf{w}}_k)(1 - U(k+1))| \\
&+ |w(\widetilde{\mathbf{w}}_k)(1 - U(k+1)) - w(\widetilde{\mathbf{w}}_k)(1 - w(k+1))|] \\
=& \frac{1}{2} \sum_{\widetilde{\mathbf{w}}_k} [U(k+1) \, |U(\widetilde{\mathbf{w}}_k) - w(\widetilde{\mathbf{w}}_k)| + w(\widetilde{\mathbf{w}}_k) \, |U(k+1) - w(k+1)| \\
&+ (1 - U(k+1)) \, |U(\widetilde{\mathbf{w}}_k) - w(\widetilde{\mathbf{w}}_k)| + w(\widetilde{\mathbf{w}}_k) \, |U(k+1) - w(k+1)|] \\
=& \frac{1}{2} \sum_{\widetilde{\mathbf{w}}_k} |U(\widetilde{\mathbf{w}}_k) - w(\widetilde{\mathbf{w}}_k)| + [U(k+1) - w(k+1)] \\
\overset{(b)}{\leq}& \sum_{e=1}^{k} [U(e) - w(e)] + [U(k+1) - w(k+1)] \\
=& \sum_{e=1}^{k+1} [U(e) - w(e)] \, ,
\end{aligned} \tag{A.31}$$

where (a) follows from the triangular inequality and (b) follows from the induction hypothesis. Hence, we have proved Equation A.30 by induction hypothesis. As we have proved above, this is sufficient to prove Lemma 5. □

Finally, we prove the following lemma:

Lemma 6. *For any two weight functions $w, U : \mathcal{E} \to [0,1]$ s.t. $U(e) \geq w(e)$ for all $e \in \mathcal{E}$, we have*

$$f(\mathcal{S}_t, U, v) - f(\mathcal{S}_t, w, v) \leq \mathbb{E}\left[\sum_{\tau=0}^{\widetilde{\tau}-1} \sum_{e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})} [U(e) - w(e)] \middle| \mathcal{S}_t \right],$$

where $\widetilde{\tau}$ is the stopping time when $\mathcal{S}^\tau = \emptyset$ or $v \in \mathcal{S}^\tau$, and the expectation is under the weight function w.

Proof. Recall that the diffusion process $(\mathcal{S}^\tau)_{\tau=0}^{\widetilde{\tau}}$ is a stochastic process. Note that by definition, if we treat the pair $(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})$ as the *state* of the diffusion process at diffusion step τ, and assume that $\mathbf{w}(e) \sim \text{Bern}\left(w(e)\right)$ are independently sampled for all $e \in \mathcal{E}_{\mathcal{S}_t, v}$, then the sequence $(\mathcal{S}^0, \mathcal{S}^{0:-1}), (\mathcal{S}^0, \mathcal{S}^{0:-1}), \ldots, (\mathcal{S}^{\widetilde{\tau}}, \mathcal{S}^{0:\widetilde{\tau}-1})$ follows a Markov chain, specifically,

- For any state $(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})$ s.t. $v \notin \mathcal{S}^\tau$ and $\mathcal{S}^\tau \neq \emptyset$, its transition probabilities to the next state $(\mathcal{S}^{\tau+1}, \mathcal{S}^{0:\tau})$ depend on $w(e)$'s for $e \in \mathcal{E}\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau}\right)$.
- Any state $(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})$ s.t. $v \in \mathcal{S}^\tau$ or $\mathcal{S}^\tau = \emptyset$ is a terminal state and the state transition terminates once visiting such a state. Recall that by definition of the stopping time $\widetilde{\tau}$, the state transition terminates at $\widetilde{\tau}$.

We define $h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w\right)$ as the "value" at state $(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1})$. Also note that the states in this Markov chain is *topologically sortable* in the sense that it will never revisit a state it visits before. Hence, we can compute $h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, U\right) - h\left(\mathcal{S}^\tau, \mathcal{S}^{0:\tau-1}, w\right)$ via a backward induction from the terminal states, based on a valid topological order. Thus, from Lemma 5, we have

$$\begin{aligned} f(\mathcal{S}_t, U, v) - f(\mathcal{S}_t, w, v) &\overset{(a)}{=} h(\mathcal{S}^0, \emptyset, U) - h(\mathcal{S}^0, \emptyset, w) \\ &\overset{(b)}{\leq} \mathbb{E}\left[\sum_{\tau=0}^{\widetilde{\tau}-1} \sum_{e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})} [U(e) - w(e)] \middle| \mathcal{S}^0 \right], \end{aligned} \tag{A.32}$$

where (a) follows from the definition of h, and (b) follows from the backward induction. Since $\mathcal{S}^0 = \mathcal{S}_t$ by definition, we have proved Lemma 6. □

Finally, we prove Theorem 9 based on Lemma 6. Recall that the favorable event at round $t-1$ is defined as

$$\xi_{t-1} = \left\{ |x_e^\mathsf{T}(\bar{\theta}_{\tau-1} - \theta^*)| \leq c\sqrt{x_e^\mathsf{T}\mathbf{M}_{\tau-1}^{-1}x_e},\ \forall e \in \mathcal{E},\ \forall \tau \leq t \right\}.$$

Also, based on Algorithm 2, we have

$$0 \leq p(e) \leq U_t(e) \leq 1, \forall e \in \mathcal{E}.$$

Thus, from Lemma 6, we have

$$f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v) \leq \mathbb{E}\left[\textstyle\sum_{\tau=0}^{\widetilde{\tau}-1} \sum_{e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})} [U_t(e) - p(e)] \middle| \mathcal{S}_t, \mathcal{H}_{t-1}\right],$$

where the expectation is based on the weight function p. Recall that $O_t(e)$ is the event that edge e is observed at round t. Recall that by definition, all edges in $\mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})$ are observed at round t (since they are going out from an influenced node in $\mathcal{S}^\tau$, see Definition 2) and belong to $\mathcal{E}_{\mathcal{S}_t, v}$, so we have

$$\begin{aligned} f(\mathcal{S}_t, U_t, v) - f(\mathcal{S}_t, p, v) &\leq \mathbb{E}\left[\sum_{\tau=0}^{\widetilde{\tau}-1} \sum_{e \in \mathcal{E}(\mathcal{S}^\tau, \mathcal{S}^{0:\tau})} [U_t(e) - p(e)] \middle| \mathcal{S}_t, \mathcal{H}_{t-1}\right] \\ &\leq \mathbb{E}\left[\sum_{e \in \mathcal{E}_{\mathcal{S}_t, v}} \mathbf{1}\left(O_t(e)\right) [U_t(e) - p(e)] \middle| \mathcal{S}_t, \mathcal{H}_{t-1}\right]. \end{aligned} \tag{A.33}$$

This completes the proof for Theorem 9.

A.1.3 Proof of Lemma 2

Proof. To simplify the exposition, we define $z_{t,e} = \sqrt{x_e^\mathsf{T}\mathbf{M}_{t-1}^{-1}x_e}$ for all $t = 1, 2 \ldots, T$ and all $e \in \mathcal{E}$, and use $\mathcal{E}_t^o$ denote the set of edges observed at round t. Recall that

$$\mathbf{M}_t = \mathbf{M}_{t-1} + \frac{1}{\sigma^2} \sum_{e \in \mathcal{E}} x_e x_e^\mathsf{T} \mathbf{1}\left\{O_t(e)\right\} = \mathbf{M}_{t-1} + \frac{1}{\sigma^2} \sum_{e \in \mathcal{E}_t^o} x_e x_e^\mathsf{T}. \tag{A.34}$$

Thus, for all (t, e) such that $e \in \mathcal{E}_t^o$ (i.e., edge e is observed at round t), we have that

$$\begin{aligned}
\det\left[\mathbf{M}_t\right] \geq & \det\left[\mathbf{M}_{t-1} + \frac{1}{\sigma^2} x_e x_e^\intercal\right] = \det\left[\mathbf{M}_{t-1}^{\frac{1}{2}}\left(\mathbf{I} + \frac{1}{\sigma^2}\mathbf{M}_{t-1}^{-\frac{1}{2}} x_e x_e^\intercal \mathbf{M}_{t-1}^{-\frac{1}{2}}\right)\mathbf{M}_{t-1}^{\frac{1}{2}}\right] \\
= & \det\left[\mathbf{M}_{t-1}\right] \det\left[\mathbf{I} + \frac{1}{\sigma^2}\mathbf{M}_{t-1}^{-\frac{1}{2}} x_e x_e^\intercal \mathbf{M}_{t-1}^{-\frac{1}{2}}\right] \\
= & \det\left[\mathbf{M}_{t-1}\right]\left(1 + \frac{1}{\sigma^2} x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e\right) = \det\left[\mathbf{M}_{t-1}\right]\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right).
\end{aligned}$$

Thus, we have

$$\left(\det\left[\mathbf{M}_t\right]\right)^{|\mathcal{E}_t^o|} \geq \left(\det\left[\mathbf{M}_{t-1}\right]\right)^{|\mathcal{E}_t^o|} \prod_{e \in \mathcal{E}_t^o}\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right).$$

Remark 1. *Notice that when the feature matrix* $\mathbf{X} = \mathbf{I}$*,* $\mathbf{M}_t$*'s are always diagonal matrices, and we have*

$$\det\left[\mathbf{M}_t\right] = \det\left[\mathbf{M}_{t-1}\right] \prod_{e \in \mathcal{E}_t^o}\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right),$$

which will lead to a tighter bound in the tabular ($\mathbf{X} = \mathbf{I}$*) case.*

Since 1) $\det\left[\mathbf{M}_t\right] \geq \det\left[\mathbf{M}_{t-1}\right]$ from Equation A.34 and 2) $|\mathcal{E}_t^o| \leq E_*$, where E_* is defined in Equation A.1 and $|\mathcal{E}_t^o| \leq E_*$ follows from its definition, we have

$$\left(\det\left[\mathbf{M}_t\right]\right)^{E_*} \geq \left(\det\left[\mathbf{M}_{t-1}\right]\right)^{E_*} \prod_{e \in \mathcal{E}_t^o}\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right).$$

Therefore, we have

$$\left(\det\left[\mathbf{M}_T\right]\right)^{E_*} \geq \left(\det\left[\mathbf{M}_0\right]\right)^{E_*} \prod_{t=1}^T \prod_{e \in \mathcal{E}_t^o}\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right) = \prod_{t=1}^T \prod_{e \in \mathcal{E}_t^o}\left(1 + \frac{z_{t,e}^2}{\sigma^2}\right),$$

since $\mathbf{M}_0 = \mathbf{I}$. On the other hand, we have that

$$\operatorname{trace}\left(\mathbf{M}_T\right) = \operatorname{trace}\left(\mathbf{I} + \frac{1}{\sigma^2}\sum_{t=1}^T \sum_{e \in \mathcal{E}_t^o} x_e x_e^\intercal\right) = d + \frac{1}{\sigma^2}\sum_{t=1}^T \sum_{e \in \mathcal{E}_t^o} \|x_e\|_2^2 \leq d + \frac{T E_*}{\sigma^2},$$

where the last inequality follows from the fact that $\|x_e\|_2 \leq 1$ and $|\mathcal{E}_t^o| \leq E_*$. From the trace-determinant inequality, we have $\frac{1}{d}\text{trace}\left(\mathbf{M}_T\right) \geq [\det(\mathbf{M}_T)]^{\frac{1}{d}}$, thus we have

$$\left[1+\frac{TE_*}{d\sigma^2}\right]^{dE_*} \geq \left[\frac{1}{d}\text{trace}\left(\mathbf{M}_T\right)\right]^{dE_*} \geq [\det(\mathbf{M}_T)]^{E_*} \geq \prod_{t=1}^{T}\prod_{e\in\mathcal{E}_t^o}\left(1+\frac{z_{t,e}^2}{\sigma^2}\right).$$

Taking the logarithm on the both sides, we have

$$dE_* \log\left[1+\frac{TE_*}{d\sigma^2}\right] \geq \sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o}\log\left(1+\frac{z_{t,e}^2}{\sigma^2}\right). \tag{A.35}$$

Notice that $z_{t,e}^2 = x_e^{\intercal}\mathbf{M}_{t-1}^{-1}x_e \leq x_e^{\intercal}\mathbf{M}_0^{-1}x_e = \|x_e\|_2^2 \leq 1$, thus we have $z_{t,e}^2 \leq \frac{\log\left(1+\frac{z_{t,e}^2}{\sigma^2}\right)}{\log\left(1+\frac{1}{\sigma^2}\right)}$. [1] Hence we have

$$\sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2 \leq \frac{1}{\log\left(1+\frac{1}{\sigma^2}\right)}\sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o}\log\left(1+\frac{z_{t,e}^2}{\sigma^2}\right) \leq \frac{dE_*\log\left[1+\frac{TE_*}{d\sigma^2}\right]}{\log\left(1+\frac{1}{\sigma^2}\right)}. \tag{A.36}$$

Remark 2. *When the feature matrix $\mathbf{X}=\mathbf{I}$, we have $d=m$,*

$$\det\left[\mathbf{M}_T\right] = \prod_{t=1}^{T}\prod_{e\in\mathcal{E}_t^o}\left(1+\frac{z_{t,e}^2}{\sigma^2}\right), \text{ and } \quad m\log\left[1+\frac{TE_*}{m\sigma^2}\right] \geq \sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o}\log\left(1+\frac{z_{t,e}^2}{\sigma^2}\right).$$

This implies that

$$\sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2 \leq \frac{m\log\left[1+\frac{T}{\sigma^2}\right]}{\log\left(1+\frac{1}{\sigma^2}\right)}, \tag{A.37}$$

since $E_* \leq m$.

Finally, from Cauchy-Schwarz inequality, we have that

$$\sum_{t=1}^{T}\sum_{e\in\mathcal{E}}\mathbf{1}\{O_t(e)\}N_{\mathcal{S}_t,e}\sqrt{x_e^{\intercal}\mathbf{M}_{t-1}^{-1}x_e} = \sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o} N_{\mathcal{S}_t,e}z_{t,e}$$

[1] Notice that for any $y \in [0,1]$, we have $y \leq \frac{\log\left(1+\frac{y}{\sigma^2}\right)}{\log\left(1+\frac{1}{\sigma^2}\right)} \triangleq \kappa(y)$. To see it, notice that $\kappa(y)$ is a strictly concave function, and $\kappa(0)=0$ and $\kappa(1)=1$.

$$\leq \sqrt{\left(\sum_{t=1}^{T}\sum_{e \in \mathcal{E}_t^o} N_{\mathcal{S}_t,e}^2\right)\left(\sum_{t=1}^{T}\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right)}$$

$$= \sqrt{\left(\sum_{t=1}^{T}\sum_{e \in \mathcal{E}} \mathbf{1}\left\{O_t(e)\right\} N_{\mathcal{S}_t,e}^2\right)\left(\sum_{t=1}^{T}\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right)}. \quad \text{(A.38)}$$

Combining this inequality with the above bounds on $\sum_{t=1}^{T}\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2$ (see Equations A.36 and A.37), we obtain the statement of the lemma. □

A.1.4 Proof of Lemma 3

Proof. We use $\mathcal{E}_t^o$ denote the set of edges observed at round t. The first observation is that we can order edges in $\mathcal{E}_t^o$ based on breadth-first search (BFS) from the source nodes $\mathcal{S}_t$, as described in Algorithm 5, where $\pi_t(\mathcal{S}_t)$ is an arbitrary conditionally deterministic order of $\mathcal{S}_t$. We say a node $u \in \mathcal{V}$ is a *downstream neighbor* of node $v \in \mathcal{V}$ if there is a directed edge (v, u). We also assume that there is a fixed order of downstream neighbors for any node $v \in \mathcal{V}$.

Algorithm 5 Breadth-First Sort of Observed Edges

```
**Input:** graph $\mathcal{G}$, $\pi_t(\mathcal{S}_t)$, and $\mathbf{w}_t$
**Initialization:** node queue queueN ← $\pi_t(\mathcal{S}_t)$, edge queue queueE ← $\emptyset$, dictionary of influenced nodes dictN ← $\mathcal{S}_t$
**while** queueN is not empty **do**
  node $v$ ← queueN.dequeue()
  **for** all downstream neighbor $u$ of $v$ **do**
    queueE.enqueue($(v, u)$)
    **if** $\mathbf{w}_t(v, u) == 1$ and $u \notin$ dictN **then**
      queueN.enqueue($u$) and dictN ← dictN $\cup \{u\}$
**Output:** edge queue queueE
```

Let $J_t = |\mathcal{E}_t^o|$. Based on Algorithm 5, we order the observed edges in $\mathcal{E}_t^o$ as $a_1^t, a_2^t, \ldots, a_{J_t}^t$. We start by defining some useful notation. For any $t = 1, 2, \ldots,$ any $j = 1, 2, \ldots, J_t$, we define

$$\eta_{t,j} = \mathbf{w}_t(a_j^t) - p(a_j^t).$$

One key observation is that $\eta_{t,j}$'s form a martingale difference sequence (MDS).[2] Moreover, $\eta_{t,j}$'s are bounded in $[-1,1]$ and hence they are conditionally sub-Gaussian with constant $R=1$. We further define that

$$\begin{aligned}\mathbf{V}_t =&\sigma^2\mathbf{M}_t = \sigma^2\mathbf{I} + \sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau} x_{a_j^\tau} x_{a_j^\tau}^\intercal, \text{ and}\\ Y_t =&\sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau} x_{a_j^\tau}\eta_{t,j} = B_t - \sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau} x_{a_j^\tau} p(a_j^t) = B_t - \left[\sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau} x_{a_j^\tau} x_{a_j^\tau}^\intercal\right]\theta^*.\end{aligned}$$

As we will see later, we define $\mathbf{V}_t$ and Y_t to use the self-normalized bound developed in (Abbasi-Yadkori et al., 2011) (see Algorithm 1 of (Abbasi-Yadkori et al., 2011)). Notice that

$$\mathbf{M}_t\overline{\theta}_t = \frac{1}{\sigma^2}B_t = \frac{1}{\sigma^2}Y_t + \frac{1}{\sigma^2}\left[\sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau} x_{a_j^\tau} x_{a_j^\tau}^\intercal\right]\theta^* = \frac{1}{\sigma^2}Y_t + \left[\mathbf{M}_t - \mathbf{I}\right]\theta^*,$$

where the last equality is based on the definition of $\mathbf{M}_t$. Hence we have

$$\overline{\theta}_t - \theta^* = \mathbf{M}_t^{-1}\left[\frac{1}{\sigma^2}Y_t - \theta^*\right].$$

Thus, for any $e \in \mathcal{E}$, we have

$$\begin{aligned}\left|\langle x_e, \overline{\theta}_t - \theta^*\rangle\right| =& \left|x_e^\intercal \mathbf{M}_t^{-1}\left[\frac{1}{\sigma^2}Y_t - \theta^*\right]\right| \leq \|x_e\|_{\mathbf{M}_t^{-1}} \|\frac{1}{\sigma^2}Y_t - \theta^*\|_{\mathbf{M}_t^{-1}}\\ \leq& \|x_e\|_{\mathbf{M}_t^{-1}}\left[\|\frac{1}{\sigma^2}Y_t\|_{\mathbf{M}_t^{-1}} + \|\theta^*\|_{\mathbf{M}_t^{-1}}\right],\end{aligned}$$

where the first inequality follows from the Cauchy-Schwarz inequality and the second inequality follows from the triangle inequality. Notice that $\|\theta^*\|_{\mathbf{M}_t^{-1}} \leq \|\theta^*\|_{\mathbf{M}_0^{-1}} = \|\theta^*\|_2$, and $\|\frac{1}{\sigma^2}Y_t\|_{\mathbf{M}_t^{-1}} = \frac{1}{\sigma}\|Y_t\|_{\mathbf{V}_t^{-1}}$ (since $\mathbf{M}_t^{-1} = \sigma^2\mathbf{V}_t^{-1}$), therefore we have

$$\left|\langle x_e, \overline{\theta}_t - \theta^*\rangle\right| \leq \|x_e\|_{\mathbf{M}_t^{-1}}\left[\frac{1}{\sigma}\|Y_t\|_{\mathbf{V}_t^{-1}} + \|\theta^*\|_2\right]. \tag{A.39}$$

[2] Notice that the notion of "time" (or a round) is indexed by the pair (t,j), and follows the lexicographical order. Based on Algorithm 5, at the beginning of round (t,j), a_j^t is conditionally deterministic and the conditional mean of $\mathbf{w}_t(a_j^t)$ is $p(a_j^t)$.

Notice that the above inequality always holds. We now provide a high-probability bound on $\|Y_t\|_{\mathbf{V}_t^{-1}}$ based on self-normalized bound proved in (Abbasi-Yadkori et al., 2011). From Theorem 1 of (Abbasi-Yadkori et al., 2011), we know that for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$\|Y_t\|_{\mathbf{V}_t^{-1}} \leq \sqrt{2\log\left(\frac{\det(\mathbf{V}_t)^{1/2}\det(\mathbf{V}_0)^{-1/2}}{\delta}\right)} \quad \forall t = 0, 1, \ldots \ .$$

Notice that $\det(\mathbf{V}_0) = \det(\sigma^2\mathbf{I}) = \sigma^{2d}$. Moreover, from the trace-determinant inequality, we have

$$[\det(\mathbf{V}_t)]^{1/d} \leq \frac{\operatorname{trace}(\mathbf{V}_t)}{d} = \sigma^2 + \frac{1}{d}\sum_{\tau=1}^{t}\sum_{j=1}^{J_\tau}\|x_{a_j^\tau}\|_2^2 \leq \sigma^2 + \frac{tE_*}{d} \leq \sigma^2 + \frac{TE_*}{d},$$

where the second inequality follows from the assumption that $\|x_{\mathsf{a}_k^t}\|_2 \leq 1$ and the fact $J_t = |\mathcal{E}_t^o| \leq E_*$, and the last inequality follows from $t \leq T$. Thus, with probability at least $1-\delta$, we have

$$\|Y_t\|_{\mathbf{V}_t^{-1}} \leq \sqrt{d\log\left(1+\frac{TE_*}{d\sigma^2}\right) + 2\log\left(\frac{1}{\delta}\right)} \quad \forall t = 0, 1, \ldots, T-1.$$

That is, with probability at least $1-\delta$, we have

$$\left|\langle x_e, \overline{\theta}_t - \theta^*\rangle\right| \leq \|x_e\|_{\mathbf{M}_t^{-1}}\left[\frac{1}{\sigma}\sqrt{d\log\left(1+\frac{TE_*}{d\sigma^2}\right) + 2\log\left(\frac{1}{\delta}\right)} + \|\theta^*\|_2\right]$$

for all $t = 0, 1, \ldots, T-1$ and $\forall e \in E$.

Recall that by the definition of event ξ_{t-1}, the above inequality implies that, for any $t = 1, 2, \ldots, T$, if

$$c \geq \frac{1}{\sigma}\sqrt{d\log\left(1+\frac{TE_*}{d\sigma^2}\right) + 2\log\left(\frac{1}{\delta}\right)} + \|\theta^*\|_2,$$

then $P(\xi_{t-1}) \geq 1-\delta$. That is, $P(\overline{\xi}_{t-1}) \leq \delta$. □

A.2 Proof for a Better Lemma 7

In this section, we prove the following lemma, which is an improved version of Lemma 2.

Lemma 7. *Under the assumption that $\|x_e\|_2^2 \leq \frac{1}{m}$ for all $e \in \mathcal{E}$, for any round $t = 1, 2, \ldots, T$, we have*

$$\sum_{t=1}^{T}\sum_{e\in\mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}\sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e} \leq \sqrt{\left(\sum_{t=1}^{T}\sum_{e\in\mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e}^2\right) \frac{d\log\left(1+\frac{T}{d\sigma^2}\right)}{\log\left(1+\frac{1}{\sigma^2}\right)}}$$

Proof. To simplify the exposition, we define $z_{t,e} = \sqrt{x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e}$ for all $t = 1, 2 \ldots, T$ and all $e \in \mathcal{E}$, and use $\mathcal{E}_t^o$ denote the set of edges observed at round t. Recall that

$$\mathbf{M}_t = \mathbf{M}_{t-1} + \frac{1}{\sigma^2}\sum_{e\in\mathcal{E}} x_e x_e^\intercal \mathbf{1}\left\{O_t(e)\right\} = \mathbf{M}_{t-1} + \frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} x_e x_e^\intercal. \tag{A.40}$$

Thus, we have

$$\begin{aligned}\det\left[\mathbf{M}_t\right] &= \det\left[\mathbf{M}_{t-1} + \frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} x_e x_e^\intercal\right] \\ &= \det\left[\mathbf{M}_{t-1}^{\frac{1}{2}}\right]\det\left[\mathbf{I} + \frac{1}{\sigma^2}\mathbf{M}_{t-1}^{-\frac{1}{2}}\left(\sum_{e\in\mathcal{E}_t^o} x_e x_e^\intercal\right)\mathbf{M}_{t-1}^{-\frac{1}{2}}\right]\det\left[\mathbf{M}_{t-1}^{\frac{1}{2}}\right] \\ &= \det\left[\mathbf{M}_{t-1}\right]\det\left[\mathbf{I} + \frac{1}{\sigma^2}\mathbf{M}_{t-1}^{-\frac{1}{2}}\left(\sum_{e\in\mathcal{E}_t^o} x_e x_e^\intercal\right)\mathbf{M}_{t-1}^{-\frac{1}{2}}\right].\end{aligned} \tag{A.41}$$

Let $\lambda_1, \ldots, \lambda_d$ be the d eigenvalues of $\frac{1}{\sigma^2}\mathbf{M}_{t-1}^{-\frac{1}{2}}\left(\sum_{e\in\mathcal{E}_t^o} x_e x_e^\intercal\right)\mathbf{M}_{t-1}^{-\frac{1}{2}}$, since the matrix is positive semi-definite, we have $\lambda_1, \ldots, \lambda_d \geq 0$. Hence we have

$$\begin{aligned}\det\left[\mathbf{M}_t\right] &= \det\left[\mathbf{M}_{t-1}\right]\prod_{i=1}^{d}(1+\lambda_i) \\ &\overset{(a)}{\geq} \det\left[\mathbf{M}_{t-1}\right]\left[1 + \sum_{i=1}^{d}\lambda_i\right]\end{aligned}$$

$$
\begin{aligned}
&= \det\left[\mathbf{M}_{t-1}\right]\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o}\operatorname{trace}\left[\mathbf{M}_{t-1}^{-\frac{1}{2}}x_e x_e^\intercal \mathbf{M}_{t-1}^{-\frac{1}{2}}\right]\right] \\
&= \det\left[\mathbf{M}_{t-1}\right]\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} x_e^\intercal \mathbf{M}_{t-1}^{-1} x_e\right] \\
&= \det\left[\mathbf{M}_{t-1}\right]\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2\right], \qquad \text{(A.42)}
\end{aligned}
$$

where (a) follows from the fact that $\lambda_1, \ldots, \lambda_d \geq 0$. Recall that $\mathbf{M}_0 = \mathbf{I}$, then we have

$$
\det\left[\mathbf{M}_T\right] \geq \prod_{t=1}^{T}\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2\right].
$$

On the other hand, we have

$$
\operatorname{trace}\left[\mathbf{M}_T\right] = d + \frac{1}{\sigma^2}\sum_{t=1}^{T}\sum_{e\in\mathcal{E}_t^o}\|x_e\|_2^2 \leq d + \frac{T}{\sigma^2},
$$

where the last inequality follows from the assumption that $\|x_e\|_2^2 \leq \frac{1}{m}$ for all $e \in \mathcal{E}$. Notice that under this assumption, we have

$$
\sum_{e\in\mathcal{E}_t^o}\|x_e\|_2^2 \leq \sum_{e\in\mathcal{E}}\|x_e\|_2^2 \leq \sum_{e\in\mathcal{E}}\frac{1}{m} = 1.
$$

Combining the above results, we have

$$
\prod_{t=1}^{T}\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2\right] \leq \det\left[\mathbf{M}_T\right] \leq \left[\frac{\operatorname{trace}\left[\mathbf{M}_T\right]}{d}\right]^d \leq \left[1+\frac{T}{d\sigma^2}\right]^d.
$$

Taking the logarithm, we have

$$
d\log\left(1+\frac{T}{d\sigma^2}\right) \geq \sum_{t=1}^{T}\log\left[1+\frac{1}{\sigma^2}\sum_{e\in\mathcal{E}_t^o} z_{t,e}^2\right]
$$

Notice that

$$\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2 \leq \sum_{e \in \mathcal{E}} x_e^\mathsf{T} \mathbf{M}_{t-1}^{-1} x_e \leq \sum_{e \in \mathcal{E}} x_e^\mathsf{T} \mathbf{M}_0^{-1} x_e = \sum_{e \in \mathcal{E}} \|x_e\|_2^2 \leq \sum_{e \in \mathcal{E}} \frac{1}{m} = 1.$$

Define auxiliary function $\kappa : \Re^+ \to \Re^+$ as $\kappa(y) = \log\left(1 + \frac{y}{\sigma^2}\right)$. Notice that $\kappa(y)$ is concave in y, and hence $\kappa(y) \geq \log\left(1 + \frac{1}{\sigma^2}\right) y$ for $y \in [0,1]$. So we have

$$d \log\left(1 + \frac{T}{d\sigma^2}\right) \geq \sum_{t=1}^{T} \log\left[1 + \frac{1}{\sigma^2} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right] = \sum_{t=1}^{T} \kappa\left(\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right) \geq \log\left(1 + \frac{1}{\sigma^2}\right) \sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2,$$

where the last inequality follows from $\sum_{e \in \mathcal{E}_t^o} z_{t,e}^2 \in [0,1]$. This implies that

$$\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2 \leq \frac{d \log\left(1 + \frac{T}{d\sigma^2}\right)}{\log\left(1 + \frac{1}{\sigma^2}\right)}.$$

Finally, from Cauchy-Schwarz inequality, we have that

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{e \in \mathcal{E}} \mathbf{1}\{O_t(e)\} N_{\mathcal{S}_t,e} \sqrt{x_e^\mathsf{T} \mathbf{M}_{t-1}^{-1} x_e} &= \sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} N_{\mathcal{S}_t,e} z_{t,e} \\
&\leq \sqrt{\left(\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} N_{\mathcal{S}_t,e}^2\right)\left(\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right)} \\
&= \sqrt{\left(\sum_{t=1}^{T} \sum_{e \in \mathcal{E}} \mathbf{1}\left\{O_t(e)\right\} N_{\mathcal{S}_t,e}^2\right)\left(\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2\right)}. \quad \text{(A.43)}
\end{aligned}$$

Combining this inequality with the above bound on $\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_t^o} z_{t,e}^2$, we obtain the statement of the lemma.

□

A.3 Laplacian Regularization

As explained in section 2.4.6, enforcing Laplacian regularization leads to the following optimization problem:

$$\widehat{\theta}_t = \arg\min_{\theta}[\sum_{j=1}^{t}\sum_{u \in \mathcal{S}_t}(y_{u,j} - \theta_u X)^2 + \lambda_2 \sum_{(u_1,u_2)\in\mathcal{E}} ||\theta_{u_1} - \theta_{u_2}||_2^2]$$

Here, the first term is the data fitting term, whereas the second term is the Laplacian regularization terms which enforces smoothness in the source node estimates. This can optimization problem can be re-written as follows:

$$\widehat{\theta}_t = \arg\min_{\theta}\left[\sum_{j=1}^{t}\sum_{u \in \mathcal{S}_t}(y_{u,j} - \theta_u X)^2 + \lambda_2 \theta^T (L \otimes I_d)\theta\right]$$

Here, $\theta \in \Re^{dn}$ is the concatenation of the n d-dimensional θ_u vectors and $A \otimes B$ refers to the Kronecker product of matrices A and B. Setting the gradient of equation A.44 to zero results in solving the following linear system:

$$[XX^T \otimes I_n + \lambda_2 L \otimes I_d]\widehat{\theta}_t = b_t \tag{A.44}$$

Here b_t corresponds to the concatenation of the n d-dimensional vectors $b_{u,t}$. This is the Sylvester equation and there exist sophisticated methods of solving it. For simplicity, we focus on the special case when the features are derived from the Laplacian eigenvectors (Section 2.4.6).

Let β_t be a diagonal matrix such that $\beta_t u, u$ refers to the number of times node u has been selected as the source. Since the Laplacian eigenvectors are orthogonal, when using Laplacian features, $XX^T \otimes I_n = \beta \otimes I_d$. We thus obtain the following system:

$$[(\beta + \lambda_2 L) \otimes I_d]\widehat{\theta}_t = b_t \tag{A.45}$$

Note that the matrix $(\beta + \lambda_2 L)$ and thus $(\beta + \lambda_2 L) \otimes I_d$ is positive semi-definite and can be solved using conjugate gradient (Hestenes and Stiefel, 1952).

For conjugate gradient, the most expensive operation is the matrix-vector multiplication $(\beta + \lambda_2 L) \otimes I_d]\mathbf{v}$ for an arbitrary vector $\mathbf{v}$. Let vec be an operation that takes a $d \times n$ matrix and stacks it column-wise converting it into a dn-dimensional vector. Let V refer to the $d \times n$ matrix obtained by partitioning the vector $\mathbf{v}$ into columns of V. Given this notation, we use the property that $(B^T \otimes A)\mathbf{v} = vec(AVB)$. This implies that the matrix-vector multiplication can then be rewritten as follows:

$$(\beta + \lambda_2 L) \otimes I_d \mathbf{v} = \text{vec}(V \left(\beta + \lambda_2 L^T\right)) \tag{A.46}$$

Since β is a diagonal matrix, $V\beta$ is an $O(dn)$ operation, whereas VL^T is an $O(dm)$ operation since there are only m non-zeros (corresponding to edges) in the Laplacian matrix. Hence the complexity of computing the mean $\widehat{\theta}_t$ is an order $O((d(m+n))\kappa)$ where κ is the number of conjugate gradient iterations. In our experiments, similar to (Vaswani et al., 2017b), we warm-start with the solution at the previous round and find that $\kappa = 5$ is enough for convergence.

Unlike independent estimation where we update the UCB estimates for only the selected nodes, when using Laplacian regularization, the upper confidence values for each reachability probability need to be recomputed in each round. Once we have an estimate of θ, calculating the mean estimates for the reachabilities for all u, v requires $O(dn^2)$ computation. This is the most expensive step when using Laplacian regularization.

We now describe how to compute the confidence intervals. For this, let $\boldsymbol{D}$ denote the diagonal of $(\beta + \lambda_2 L)^{-1}$. The UCB value $z_{u,v,t}$ can then be computed as:

$$z_{u,v,t} = \sqrt{\boldsymbol{D}_u}||x_v||_2 \tag{A.47}$$

The ℓ_2 norms for all the target nodes v can be pre-computed. If we maintain the $\boldsymbol{D}$ vector, the confidence intervals for all pairs can be computed in $O(n^2)$ time.

Note that $\boldsymbol{D}_t$ requires $O(n)$ storage and can be updated across rounds in $O(K)$ time using the Sherman Morrison formula. Specifically, if $\boldsymbol{D}_{u,t}$ refers to the u^{th} element in the vector $\boldsymbol{D}_t$, then

$$\boldsymbol{D}_{u,t+1} = \begin{cases} \dfrac{\boldsymbol{D}_{u,t}}{(1 + \boldsymbol{D}_{u,t})}, & \text{if} u \in \mathcal{S}_t \\ \boldsymbol{D}_{u,t}, & \text{otherwise} \end{cases}$$

Hence, the total complexity of implementing Laplacian regularization is $O(dn^2)$. We need to store the θ vector, the Laplacian and the diagonal vectors β and $\boldsymbol{D}$. Hence, the total memory requirement is $O(dn + m)$.

A.4 Proof of theorem 2

Proof. Theorem 2 can be proved based on the definitions of monotonicity and submodularity. Note that from Assumption 1, for any seed set $\mathcal{S} \in \mathcal{C}$, any seed node $u \in \mathcal{S}$, and any target node $v \in \mathcal{V}$, we have $f(\{u\}, v) \leq f(\mathcal{S}, v)$, which implies that

$$\widetilde{f}(\mathcal{S}, v, p^*) = \max_{u \in \mathcal{S}} f(\{u\}, v) \leq f(\mathcal{S}, v),$$

hence

$$\widetilde{f}(\mathcal{S}, p^*) = \sum_{v \in \mathcal{V}} \widetilde{f}(\mathcal{S}, v, p^*) \leq \sum_{v \in \mathcal{V}} f(\mathcal{S}, v) = f(\mathcal{S}).$$

This proves the first part of theorem 2.

We now prove the second part of the theorem. First, note that from the first part, we have

$$\widetilde{f}(\widetilde{\mathcal{S}}, p^*) \leq f(\widetilde{\mathcal{S}}) \leq f(\mathcal{S}^*),$$

where the first inequality follows from the first part of this theorem, and the second inequality follows from the definition of $\mathcal{S}^*$. Thus, we have $\rho \leq 1$. To prove that $\rho \geq 1/K$, we assume that $\mathcal{S} = \{u_1, u_2, \ldots, u_K\}$, and define $\mathcal{S}_k = \{u_1, u_2, \ldots, u_k\}$ for $k = 1, 2, \ldots, K$. Thus, for any $\mathcal{S} \subseteq \mathcal{V}$ with $|\mathcal{S}| = K$, we have

$$\begin{aligned} f(\mathcal{S}) &= f(\mathcal{S}_1) + \sum_{k=1}^{K-1} [f(\mathcal{S}_{k+1}) - f(\mathcal{S}_k)] \\ &\leq \sum_{k=1}^{K} f(\{u_k\}) = \sum_{k=1}^{K} \sum_{v \in \mathcal{V}} f(\{u_k\}, v) \\ &\leq \sum_{v \in \mathcal{V}} K \max_{u \in \mathcal{S}} f(\{u\}, v) = K \sum_{v \in \mathcal{V}} \widetilde{f}(\mathcal{S}, v, p^*) = K \widetilde{f}(\mathcal{S}, p^*), \end{aligned}$$

where the first inequality follows from the submodularity of $f(\cdot)$. Thus we have

$$f(\mathcal{S}^*) \leq K \widetilde{f}(\mathcal{S}^*, p^*) \leq K \widetilde{f}(\widetilde{\mathcal{S}}, p^*),$$

where the second inequality follows from the definition of $\widetilde{\mathcal{S}}$. This implies that $\rho \geq 1/K$. □

A.5 Proof of theorem 3

We start by defining some useful notations. We use $\mathcal{H}_t$ to denote the "history" by the end of time t. For any node pair $(u, v) \in \mathcal{V} \times \mathcal{V}$ and any time t, we define the upper confidence bound (UCB) $U_t(u, v)$ and the lower confidence bound (LCB) $L_t(u, v)$ respectively as

$$\begin{aligned} U_t(u,v) &= \mathrm{Proj}_{[0,1]}\left(\langle \widehat{\theta}_{u,t-1}, \mathbf{x}_v \rangle + c\sqrt{\mathbf{x}_v^T \Sigma_{u,t-1}^{-1} \mathbf{x}_v}\right) \\ L_t(u,v) &= \mathrm{Proj}_{[0,1]}\left(\langle \widehat{\theta}_{u,t-1}, \mathbf{x}_v \rangle - c\sqrt{\mathbf{x}_v^T \Sigma_{u,t-1}^{-1} \mathbf{x}_v}\right) \end{aligned} \tag{A.48}$$

Notice that U_t is the same as the UCB estimate $\overline{q}$ defined in algorithm 3. Moreover, we define the "good event" $\mathcal{F}$ as

$$\mathcal{F} = \left\{ |x_v^T(\widehat{\theta}_{u,t-1} - \theta_u^*)| \le c\sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v},\ \forall u, v \in \mathcal{V},\ \forall t \le T \right\}, \tag{A.49}$$

and the "bad event" $\overline{\mathcal{F}}$ as the complement of $\mathcal{F}$.

A.5.1 Regret Decomposition

Recall that the realized scaled regret at time t is $R_t^{\rho\alpha} = f(\mathcal{S}^*) - \frac{1}{\rho\alpha} f(\mathcal{S}_t)$, thus we have

$$R_t^{\rho\alpha} = f(\mathcal{S}^*) - \frac{1}{\rho\alpha} f(\mathcal{S}_t) \overset{(a)}{=} \frac{1}{\rho}\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\rho\alpha} f(\mathcal{S}_t) \overset{(b)}{\le} \frac{1}{\rho}\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\rho\alpha}\widetilde{f}(\mathcal{S}_t, p^*), \tag{A.50}$$

where equality (a) follows from the definition of ρ (i.e. ρ is defined as $\rho = \widetilde{f}(\widetilde{\mathcal{S}}, p^*)/f(\mathcal{S}^*)$), and inequality (b) follows from $\widetilde{f}(\mathcal{S}_t, p^*) \le f(\mathcal{S}_t)$ (see theorem 2). Thus, we have

$$\begin{aligned} R^{\rho\alpha}(T) =& \mathbb{E}\left[\sum_{t=1}^T R_t^{\rho\alpha}\right] \\ \le& \frac{1}{\rho}\mathbb{E}\left\{\sum_{t=1}^T \left[\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \widetilde{f}(\mathcal{S}_t, p^*)/\alpha\right]\right\} \\ =& \frac{P(\mathcal{F})}{\rho}\mathbb{E}\left\{\sum_{t=1}^T \left[\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \widetilde{f}(\mathcal{S}_t, p^*)/\alpha\right]\middle|\mathcal{F}\right\} + \frac{P(\overline{\mathcal{F}})}{\rho}\mathbb{E}\left\{\sum_{t=1}^T \left[\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \widetilde{f}(\mathcal{S}_t, p^*)/\alpha\right]\middle|\overline{\mathcal{F}}\right\} \end{aligned}$$

$$\leq \frac{1}{\rho}\mathbb{E}\left\{ \sum_{t=1}^{T} \left[\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \widetilde{f}(\mathcal{S}_t, p^*)/\alpha \right] \middle| \mathcal{F} \right\} + \frac{P(\overline{\mathcal{F}})}{\rho} nT, \tag{A.51}$$

where the last inequality follows from the naive bounds $P(\mathcal{F}) \leq 1$ and $\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \widetilde{f}(\mathcal{S}_t, p^*)/\alpha \leq n$. Notice that under "good" event $\mathcal{F}$, we have

$$L_t(u,v) \leq p^*_{uv} = x_v^T \theta_u^* \leq U_t(u,v) \tag{A.52}$$

for all node pair (u,v) and for all time $t \leq T$. Thus, we have $\widetilde{f}(\mathcal{S}, L_t) \leq \widetilde{f}(\mathcal{S}, p^*) \leq \widetilde{f}(\mathcal{S}, U_t)$ for all $\mathcal{S}$ and $t \leq T$ under event $\mathcal{F}$. So under event $\mathcal{F}$, we have

$$\widetilde{f}(\mathcal{S}_t, L_t) \overset{(a)}{\leq} \widetilde{f}(\mathcal{S}_t, p^*) \overset{(b)}{\leq} \widetilde{f}(\widetilde{\mathcal{S}}, p^*) \overset{(c)}{\leq} \widetilde{f}(\widetilde{\mathcal{S}}, U_t) \leq \max_{\mathcal{S} \in \mathcal{C}} \widetilde{f}(\mathcal{S}, U_t) \overset{(d)}{\leq} \frac{1}{\alpha} \widetilde{f}(\mathcal{S}_t, U_t)$$

for all $t \leq T$, where inequalities (a) and (c) follow from (A.52), inequality (b) follows from $\widetilde{\mathcal{S}} \in \arg\max_{\mathcal{S}\in\mathcal{C}} \widetilde{f}(\mathcal{S}, p^*)$, and inequality (d) follows from the fact that `ORACLE` is an α-approximation algorithm. Specifically, the fact that `ORACLE` is an α-approximation algorithm implies that $\widetilde{f}(\mathcal{S}_t, U_t) \geq \alpha \max_{\mathcal{S}\in\mathcal{C}} \widetilde{f}(\mathcal{S}, U_t)$.

Consequently, under event $\mathcal{F}$, we have

$$\begin{aligned}
\widetilde{f}(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\alpha}\widetilde{f}(\mathcal{S}_t, p^*) &\leq \frac{1}{\alpha}\widetilde{f}(\mathcal{S}_t, U_t) - \frac{1}{\alpha}\widetilde{f}(\mathcal{S}_t, L_t) \\
&= \frac{1}{\alpha}\sum_{v\in\mathcal{V}} \left[\max_{u\in\mathcal{S}_t} U_t(u,v) - \max_{u\in\mathcal{S}_t} L_t(u,v) \right] \\
&\leq \frac{1}{\alpha}\sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{S}_t} \left[U_t(u,v) - L_t(u,v) \right] \\
&\leq \sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{S}_t} \frac{2c}{\alpha}\sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v}.
\end{aligned} \tag{A.53}$$

So we have

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{ \sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}} \sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v} \middle| \mathcal{F} \right\} + \frac{P(\overline{\mathcal{F}})}{\rho} nT. \tag{A.54}$$

In the remainder of this section, we will provide a worst-case bound on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v}$

(see appendix A.5.2) and a bound on the probability of "bad event" $P(\overline{\mathcal{F}})$ (see appendix A.5.3).

A.5.2 Worst-Case Bound on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$

Notice that

$$\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}=\sum_{u\in\mathcal{V}}\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$$

For each $u\in\mathcal{V}$, we define $K_u=\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]$ as the number of times at which u is chosen as a source node, then we have the following lemma:

Lemma 8. *For all $u\in\mathcal{V}$, we have*

$$\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\leq\sqrt{nK_u}\sqrt{\frac{dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}.$$

Moreover, when $X=I$, we have

$$\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\leq\sqrt{nK_u}\sqrt{\frac{n\log\left(1+\frac{K_u}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}.$$

Proof. To simplify the exposition, we use Σ_t to denote $\Sigma_{u,t}$, and define $z_{t,v}=\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$ for all $t\leq T$ and all $v\in\mathcal{V}$. Recall that

$$\Sigma_t=\Sigma_{t-1}+\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}XX^T=\Sigma_{t-1}+\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}x_vx_v^T.$$

Note that if $u\notin\mathcal{S}_t$, $\Sigma_t=\Sigma_{t-1}$. If $u\in\mathcal{S}_t$, then for any $v\in\mathcal{V}$, we have

$$\begin{aligned}\det\left[\Sigma_t\right]&\geq\det\left[\Sigma_{t-1}+\frac{1}{\sigma^2}x_vx_v^T\right]\\&=\det\left[\Sigma_{t-1}^{\frac{1}{2}}\left(I+\frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}x_vx_v^T\Sigma_{t-1}^{-\frac{1}{2}}\right)\Sigma_{t-1}^{\frac{1}{2}}\right]\\&=\det\left[\Sigma_{t-1}\right]\det\left[I+\frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}x_vx_v^T\Sigma_{t-1}^{-\frac{1}{2}}\right]\end{aligned}$$

$$= \det\left[\Sigma_{t-1}\right]\left(1+\frac{1}{\sigma^2}x_v^T\Sigma_{t-1}^{-1}x_v\right) = \det\left[\Sigma_{t-1}\right]\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right).$$

Hence, we have

$$\det\left[\Sigma_t\right]^n \geq \det\left[\Sigma_{t-1}\right]^n \prod_{v\in\mathcal{V}}\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \tag{A.55}$$

Note that the above inequality holds for any X. However, if $X = I$, then all Σ_t's are diagonal and we have

$$\det\left[\Sigma_t\right] = \det\left[\Sigma_{t-1}\right] \prod_{v\in\mathcal{V}}\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \tag{A.56}$$

As we will show later, this leads to a tighter regret bound in the tabular ($X = I$) case.

Let's continue our analysis for general X. The above results imply that

$$n\log\left(\det\left[\Sigma_t\right]\right) \geq n\log\left(\det\left[\Sigma_{t-1}\right]\right) + \mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right)$$

and hence

$$\begin{aligned} n\log\left(\det\left[\Sigma_T\right]\right) \geq & \, n\log\left(\det\left[\Sigma_0\right]\right) + \sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right) \\ = & \, nd\log(\lambda) + \sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \end{aligned} \tag{A.57}$$

On the other hand, we have that

$$\begin{aligned} \mathrm{Tr}\left[\Sigma_T\right] = & \, \mathrm{Tr}\left[\Sigma_0 + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}x_v x_v^T\right] \\ = & \, \mathrm{Tr}\left[\Sigma_0\right] + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}\mathrm{Tr}\left[x_v x_v^T\right] \\ = & \, \lambda d + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}\|x_v\|^2 \leq \lambda d + \frac{nK_u}{\sigma^2}, \end{aligned} \tag{A.58}$$

where the last inequality follows from the assumption that $\|x_v\| \leq 1$ and the definition of K_u. From the trace-determinant inequality, we have $\frac{1}{d}\operatorname{Tr}[\Sigma_T] \geq \det[\Sigma_T]^{\frac{1}{d}}$. Thus, we have

$$dn\log\left(\lambda + \frac{nK_u}{d\sigma^2}\right) \geq dn\log\left(\frac{1}{d}\operatorname{Tr}[\Sigma_T]\right) \geq n\log\left(\det[\Sigma_T]\right) \geq dn\log(\lambda) + \sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1 + \frac{z_{t-1,v}^2}{\sigma^2}\right).$$

That is

$$\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1 + \frac{z_{t-1,v}^2}{\sigma^2}\right) \leq dn\log\left(1 + \frac{nK_u}{d\lambda\sigma^2}\right)$$

Notice that $z_{t-1,v}^2 = x_v^T\Sigma_{t-1}^{-1}x_v \leq x_v^T\Sigma_0^{-1}x_v = \frac{\|x_v\|^2}{\lambda} \leq \frac{1}{\lambda}$. Moreover, for all $y \in [0, 1/\lambda]$, we have $\log\left(1 + \frac{y}{\sigma^2}\right) \geq \lambda\log\left(1 + \frac{1}{\lambda\sigma^2}\right)y$ based on the concavity of $\log(\cdot)$. Thus, we have

$$\lambda\log\left(1 + \frac{1}{\lambda\sigma^2}\right)\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v}^2 \leq dn\log\left(1 + \frac{nK_u}{d\lambda\sigma^2}\right).$$

Finally, from Cauchy-Schwarz inequality, we have that

$$\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v}^2}.$$

Combining the above results, we have

$$\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\frac{dn\log\left(1 + \frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda\log\left(1 + \frac{1}{\lambda\sigma^2}\right)}}. \tag{A.59}$$

This concludes the proof for general X. Based on (A.56), the analysis for the tabular ($X = I$) case is similar, and we omit the detailed analysis. In the tabular case, we have

$$\sum_{t=1}^{T}\mathbf{1}\left(u \in \mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\frac{n\log\left(1 + \frac{K_u}{\lambda\sigma^2}\right)}{\lambda\log\left(1 + \frac{1}{\lambda\sigma^2}\right)}}. \tag{A.60}$$

□

We now develop a worst-case bound. Notice that for general X, we have

$$\begin{aligned}
\sum_{u \in \mathcal{V}} \sum_{t=1}^{T} \mathbf{1}\left[u \in \mathcal{S}_t\right] \sum_{v \in \mathcal{V}} \sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v} &\leq \sum_{u \in \mathcal{V}} \sqrt{nK_u} \sqrt{\frac{dn \log\left(1 + \frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} \\
&\overset{(a)}{\leq} n \sqrt{\frac{d \log\left(1 + \frac{nT}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} \sum_{u \in \mathcal{V}} \sqrt{K_u} \\
&\overset{(b)}{\leq} n \sqrt{\frac{d \log\left(1 + \frac{nT}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} \sqrt{n} \sqrt{\sum_{u \in \mathcal{V}} K_u} \\
&\overset{(c)}{=} n^{\frac{3}{2}} \sqrt{\frac{dKT \log\left(1 + \frac{nT}{d\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}},
\end{aligned} \tag{A.61}$$

where inequality (a) follows from the naive bound $K_u \leq T$, inequality (b) follows from Cauchy-Schwarz inequality, and equality (c) follows from $\sum_{u \in \mathcal{V}} K_u = KT$. Similarly, for the special case with $X = I$, we have

$$\sum_{u \in \mathcal{V}} \sum_{t=1}^{T} \mathbf{1}\left[u \in \mathcal{S}_t\right] \sum_{v \in \mathcal{V}} \sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v} \leq \sum_{u \in \mathcal{V}} \sqrt{nK_u} \sqrt{\frac{n \log\left(1 + \frac{K_u}{\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}} \leq n^{\frac{3}{2}} \sqrt{\frac{KT \log\left(1 + \frac{T}{\lambda\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}}. \tag{A.62}$$

This concludes the derivation of a worst-case bound.

A.5.3 Bound on $P\left(\overline{\mathcal{F}}\right)$

We now derive a bound on $P\left(\overline{\mathcal{F}}\right)$ based on the "Self-Normalized Bound for Matrix-Valued Martingales" developed in appendix A.6 (see theorem 10). Before proceeding, we define $\mathcal{F}_u$ for all $u \in \mathcal{V}$ as

$$\mathcal{F}_u = \left\{ |x_v^T(\widehat{\theta}_{u,t-1} - \theta_u^*)| \leq c\sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v}, \, \forall v \in \mathcal{V}, \, \forall t \leq T \right\}, \tag{A.63}$$

and the $\overline{\mathcal{F}}_u$ as the complement of $\mathcal{F}_u$. Note that by definition, $\overline{\mathcal{F}} = \bigcup_{u \in \mathcal{V}} \overline{\mathcal{F}}_u$. Hence, we first develop a bound on $P\left(\overline{\mathcal{F}}_u\right)$, then we develop a bound on $P\left(\overline{\mathcal{F}}\right)$ based on union bound.

Lemma 9. *For all $u \in \mathcal{V}$, all $\sigma, \lambda > 0$, all $\delta \in (0,1)$, and all*

$$c \geq \frac{1}{\sigma}\sqrt{dn \log\left(1 + \frac{nT}{\sigma^2 \lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda}\|\theta_u^*\|_2$$

we have $P\left(\overline{\mathcal{F}}_u\right) \leq \delta$.

Proof. To simplify the expositions, we omit the subscript u in this proof. For instance, we use θ^*, Σ_t, $\mathbf{y}_t$ and $\mathbf{b}_t$ to respectively denote θ_u^*, $\Sigma_{u,t}$, $\mathbf{y}_{u,t}$ and $\mathbf{b}_{u,t}$. We also use $\mathcal{H}_t$ to denote the "history" by the end of time t, and hence $\{\mathcal{H}_t\}_{t=0}^{\infty}$ is a filtration. Notice that U_t is $\mathcal{H}_{t-1}$-adaptive, and hence $\mathcal{S}_t$ and $\mathbf{1}[u \in \mathcal{S}_t]$ are also $\mathcal{H}_{t-1}$-adaptive. We define

$$\eta_t = \begin{cases} \mathbf{y}_t - X^T\theta^* & \text{if } u \in \mathcal{S}_t \\ 0 & \text{otherwise} \end{cases} \in \Re^n \quad \text{and} \quad X_t = \begin{cases} X & \text{if } u \in \mathcal{S}_t \\ 0 & \text{otherwise} \end{cases} \in \Re^{d \times n} \tag{A.64}$$

Note that X_t is $\mathcal{H}_{t-1}$-adaptive, and η_t is $\mathcal{H}_t$-adaptive. Moreover, $\|\eta_t\|_\infty \leq 1$ always holds, and $\mathbb{E}[\eta_t|\mathcal{H}_{t-1}] = 0$. To simplify the expositions, we further define $\mathbf{y}_t = 0$ for all t s.t. $u \notin \mathcal{S}_t$. Note that with this definition, we have $\eta_t = \mathbf{y}_t - X_t^T\theta^*$ for all t. We further define

$$\begin{aligned} \overline{V}_t &= n\sigma^2\Sigma_t = n\sigma^2\lambda I + n\sum_{s=1}^{t} X_s X_s^T \\ \overline{S}_t &= \sum_{s=1}^{t} X_s\eta_s = \sum_{s=1}^{t} X_s\left[\mathbf{y}_s - X_s^T\theta^*\right] = \mathbf{b}_t - \sigma^2\left[\Sigma_t - \lambda I\right]\theta^* \end{aligned} \tag{A.65}$$

Thus, we have $\Sigma_t\widehat{\theta}_t = \sigma^{-2}\mathbf{b}_t = \sigma^{-2}\overline{S}_t + [\Sigma_t - \lambda I]\theta^*$, which implies

$$\widehat{\theta}_t - \theta^* = \Sigma_t^{-1}\left[\sigma^{-2}\overline{S}_t - \lambda\theta^*\right]. \tag{A.66}$$

Consequently, for any $v \in \mathcal{V}$, we have

$$\begin{aligned} \left|x_v^T\left(\widehat{\theta}_t - \theta^*\right)\right| &= \left|x_v^T\Sigma_t^{-1}\left[\sigma^{-2}\overline{S}_t - \lambda\theta^*\right]\right| \leq \sqrt{x_v^T\Sigma_t^{-1}x_v}\|\sigma^{-2}\overline{S}_t - \lambda\theta^*\|_{\Sigma_t^{-1}} \\ &\leq \sqrt{x_v^T\Sigma_t^{-1}x_v}\left[\|\sigma^{-2}\overline{S}_t\|_{\Sigma_t^{-1}} + \|\lambda\theta^*\|_{\Sigma_t^{-1}}\right], \end{aligned} \tag{A.67}$$

where the first inequality follows from Cauchy-Schwarz inequality and the second inequality follows from triangular inequality. Note that $\|\lambda\theta^*\|_{\Sigma_t^{-1}} = \lambda\|\theta^*\|_{\Sigma_t^{-1}} \leq \lambda\|\theta^*\|_{\Sigma_0^{-1}} = \sqrt{\lambda}\|\theta^*\|_2$. On the other hand, since $\Sigma_t^{-1} = n\sigma^2\overline{V}_t^{-1}$, we have $\|\sigma^{-2}\overline{S}_t\|_{\Sigma_t^{-1}} = \frac{\sqrt{n}}{\sigma}\|\overline{S}_t\|_{\overline{V}_t^{-1}}$. Thus, we have

$$\left|x_v^T\left(\widehat{\theta}_t - \theta^*\right)\right| \leq \sqrt{x_v^T\Sigma_t^{-1}x_v}\left[\frac{\sqrt{n}}{\sigma}\|\overline{S}_t\|_{\overline{V}_t^{-1}} + \sqrt{\lambda}\|\theta^*\|_2\right]. \tag{A.68}$$

From theorem 10, we know with probability at least $1-\delta$, for all $t \leq T$, we have

$$\|S_t\|^2_{\overline{V}_t^{-1}} \leq 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right) \leq 2\log\left(\frac{\det\left(\overline{V}_T\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right),$$

where $V = n\sigma^2\lambda I$. Note that from the trace-determinant inequality, we have

$$\det\left[\overline{V}_T\right]^{\frac{1}{d}} \leq \frac{\mathrm{Tr}\left[\overline{V}_T\right]}{d} \leq \frac{n\sigma^2\lambda d + n^2T}{d},$$

where the last inequality follows from $\mathrm{Tr}\left[X_tX_t^T\right] \leq n$ for all t. Note that $\det\left[V\right] = \left[n\sigma^2\lambda\right]^d$, with a little bit algebra, we have

$$\|S_t\|_{\overline{V}_t^{-1}} \leq \sqrt{d\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} \quad \forall t \leq T$$

with probability at least $1-\delta$. Thus, if

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda}\|\theta^*\|_2,$$

then $\mathcal{F}_u$ holds with probability at least $1-\delta$. This concludes the proof of this lemma. □

Hence, from the union bound, we have the following lemma:

Lemma 10. *For all $\sigma, \lambda > 0$, all $\delta \in (0,1)$, and all*

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{n}{\delta}\right)} + \sqrt{\lambda}\max_{u\in\mathcal{V}}\|\theta_u^*\|_2 \tag{A.69}$$

we have $P\left(\overline{\mathcal{F}}\right) \leq \delta$.

Proof. This lemma follows directly from the union bound. Note that for all c satisfying

Equation A.69, we have $P\left(\overline{\mathcal{F}}_u\right) \leq \frac{\delta}{n}$ for all $u \in \mathcal{V}$, which implies $P\left(\overline{\mathcal{F}}\right) = P\left(\bigcup_{u\in\mathcal{V}} \overline{\mathcal{F}}_u\right) \leq \sum_{u\in\mathcal{V}} P\left(\overline{\mathcal{F}}_u\right) \leq \delta$. □

A.5.4 Conclude the Proof

Note that if we choose

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(n^2T\right)} + \sqrt{\lambda}\max_{u\in\mathcal{V}}\|\theta_u^*\|_2, \tag{A.70}$$

we have $P\left(\overline{\mathcal{F}}\right) \leq \frac{1}{nT}$. Hence for general X, we have

$$\begin{aligned} R^{\rho\alpha}(T) &\leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\middle|\mathcal{F}\right\} + \frac{1}{\rho} \\ &\leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{dKT\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}} + \frac{1}{\rho}. \end{aligned} \tag{A.71}$$

Note that with $c = \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(n^2T\right)} + \sqrt{\lambda}\max_{u\in\mathcal{V}}\|\theta_u^*\|_2$, this regret bound is $\widetilde{O}\left(\frac{n^2d\sqrt{KT}}{\rho\alpha}\right)$. Similarly, for the special case $X = I$, we have

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{KT\log\left(1+\frac{T}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}} + \frac{1}{\rho}. \tag{A.72}$$

Note that with $c = \frac{n}{\sigma}\sqrt{\log\left(1+\frac{T}{\sigma^2\lambda}\right) + 2\log\left(n^2T\right)} + \sqrt{\lambda}\max_{u\in\mathcal{V}}\|\theta_u^*\|_2 \leq \frac{n}{\sigma}\sqrt{\log\left(1+\frac{T}{\sigma^2\lambda}\right) + 2\log\left(n^2T\right)} + \sqrt{\lambda n}$, this regret bound is $\widetilde{O}\left(\frac{n^{\frac{5}{2}}\sqrt{KT}}{\rho\alpha}\right)$.

A.6 Self-Normalized Bound for Matrix-Valued Martingales

In this section, we derive a "self-normalized bound" for matrix-valued Martingales. This result is a natural generalization of Theorem 1 in Abbasi-Yadkori et al. (2011).

Theorem 10. *(Self-Normalized Bound for Matrix-Valued Martingales) Let $\{\mathcal{H}_t\}_{t=0}^{\infty}$ be a filtration, and $\{\eta_t\}_{t=1}^{\infty}$ be a $\Re^K$-valued Martingale difference sequence with respect to $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Specifically, for all t, η_t is $\mathcal{H}_t$-measurable and satisfies (1) $\mathbb{E}\left[\eta_t|\mathcal{H}_{t-1}\right] = 0$ and (2) $\|\eta_t\|_\infty \leq 1$*

with probability 1 conditioning on $\mathcal{H}_{t-1}$. Let $\{X_t\}_{t=1}^{\infty}$ be a $\Re^{d\times K}$-valued stochastic process such that X_t is $\mathcal{H}_{t-1}$ measurable. Assume that $V \in \Re^{d\times d}$ is a positive-definite matrix. For any $t \geq 0$, define

$$\overline{V}_t = V + K\sum_{s=1}^{t} X_s X_s^T \qquad S_t = \sum_{s=1}^{t} X_s \eta_s. \tag{A.73}$$

Then, for any $\delta > 0$, with probability at least $1-\delta$, we have

$$\|S_t\|^2_{\overline{V}_t^{-1}} \leq 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right) \quad \forall t \geq 0. \tag{A.74}$$

We first define some useful notations. Similarly as Abbasi-Yadkori et al. (2011), for any $\lambda \in \Re^d$ and any t, we define D_t^λ as

$$D_t^\lambda = \exp\left(\lambda^T X_t \eta_t - \frac{K}{2}\|X_t^T\lambda\|_2^2\right), \tag{A.75}$$

and $M_t^\lambda = \prod_{s=1}^{t} D_s^\lambda$ with convention $M_0^\lambda = 1$. Note that both D_t^λ and M_t^λ are $\mathcal{H}_t$-measurable, and $\left\{M_t^\lambda\right\}_{t=0}^{\infty}$ is a supermartingale with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. To see it, notice that conditioning on $\mathcal{H}_{t-1}$, we have

$$\lambda^T X_t \eta_t = (X_t^T\lambda)^T\eta_t \leq \|X_t^T\lambda\|_1\|\eta_t\|_\infty \leq \|X_t^T\lambda\|_1 \leq \sqrt{K}\|X_t^T\lambda\|_2$$

with probability 1. This implies that $\lambda^T X_t \eta_t$ is conditionally $\sqrt{K}\|X_t^T\lambda\|_2$-subGaussian. Thus, we have

$$\mathbb{E}\left[D_t^\lambda \middle| \mathcal{H}_{t-1}\right] = \mathbb{E}\left[\exp\left(\lambda^T X_t\eta_t\right)\middle|\mathcal{H}_{t-1}\right]\exp\left(-\frac{K}{2}\|X_t^T\lambda\|_2^2\right) \leq \exp\left(\frac{K}{2}\|X_t^T\lambda\|_2^2 - \frac{K}{2}\|X_t^T\lambda\|_2^2\right) = 1.$$

Thus,

$$\mathbb{E}\left[M_t^\lambda \middle| \mathcal{H}_{t-1}\right] = M_{t-1}^\lambda \mathbb{E}\left[D_t^\lambda\middle|\mathcal{H}_{t-1}\right] \leq M_{t-1}^\lambda.$$

So $\left\{M_t^\lambda\right\}_{t=0}^{\infty}$ is a supermartingale with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Then, following Lemma 8 of Abbasi-Yadkori et al. (2011), we have the following lemma:

Lemma 11. *Let τ be a stopping time with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Then for any $\lambda \in \Re^d$, M_τ^λ is almost surely well-defined and $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$.*

Proof. First, we argue that M_τ^λ is almost surely well-defined. By Doob's convergence theorem for nonnegative supermartingales, $M_\infty^\lambda = \lim_{t\to\infty} M_t^\lambda$ is almost surely well-defined. Hence M_τ^λ is indeed well-defined independent of $\tau < \infty$ or not. Next, we show that $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$. Let $Q_t^\lambda = M_{\min\{\tau,t\}}^\lambda$ be a stopped version of $\left\{M_t^\lambda\right\}_{t=1}^\infty$. By Fatou's Lemma, we have $\mathbb{E}\left[M_\tau^\lambda\right] = \mathbb{E}\left[\liminf_{t\to\infty} Q_t^\lambda\right] \leq \liminf_{t\to\infty} \mathbb{E}\left[Q_t^\lambda\right] \leq 1$. □

The following results follow from Lemma 9 of Abbasi-Yadkori et al. (2011), which uses the "method of mixtures" technique. Let Λ be a Gaussian random vector in $\Re^d$ with mean 0 and covariance matrix V^{-1}, and independent of all the other random variables. Let $\mathcal{H}_\infty$ be the tail σ-algebra of the filtration, i.e. the σ-algebra generated by the union of all events in the filtration. We further define $M_t = \mathbb{E}\left[M_t^\Lambda \middle| \mathcal{H}_\infty\right]$ for all $t = 0, 1, \ldots$ and $t = \infty$. Note that M_∞ is almost surely well-defined since M_∞^λ is almost surely well-defined.

Let τ be a stopping time with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^\infty$. Note that M_τ is almost surely well-defined since M_∞ is almost surely well-defined. Since $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$ from Lemma 11, we have

$$\mathbb{E}\left[M_\tau\right] = \mathbb{E}\left[M_\tau^\Lambda\right] = \mathbb{E}\left[\mathbb{E}\left[M_\tau^\Lambda \middle| \Lambda\right]\right] \leq 1.$$

The following lemma follows directly from the proof for Lemma 9 of Abbasi-Yadkori et al. (2011), which can be derived by algebra. The proof is omitted here.

Lemma 12. *For all finite $t = 0, 1, \ldots$, we have*

$$M_t = \left(\frac{\det(V)}{\det(\overline{V}_t)}\right)^{1/2} \exp\left(\frac{1}{2}\|S_t\|_{\overline{V}_t^{-1}}\right). \tag{A.76}$$

Note that Lemma 12 implies that for finite t, $\|S_t\|^2_{\overline{V}_t^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det(V)^{-1/2}}{\delta}\right)$ and $M_t > \frac{1}{\delta}$ are equivalent. Consequently, for any stopping time τ, the event

$$\left\{\tau < \infty,\, \|S_\tau\|^2_{\overline{V}_\tau^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right)\right\}$$

is equivalent to $\left\{\tau < \infty,\, M_\tau > \frac{1}{\delta}\right\}$. Finally, we prove Theorem 10:

Proof. We define the "bad event" at time $t = 0, 1, \dots$ as:

$$B_t(\delta) = \left\{ \|S_t\|^2_{\overline{V}_t^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta} \right) \right\}.$$

We are interested in bounding the probability of the "bad event" $\bigcup_{t=1}^{\infty} B_t(\delta)$. Let Ω denote the sample space, for any outcome $\omega \in \Omega$, we define $\tau(\omega) = \min\{t \geq 0 : \omega \in B_t(\delta)\}$, with the convention that $\min \emptyset = +\infty$. Thus, τ is a stopping time. Notice that $\bigcup_{t=1}^{\infty} B_t(\delta) = \{\tau < \infty\}$. Moreover, if $\tau < \infty$, then by definition of τ, we have $\|S_\tau\|^2_{\overline{V}_\tau^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det(V)^{-1/2}}{\delta} \right)$, which is equivalent to $M_\tau > \frac{1}{\delta}$ as discussed above. Thus we have

$$\begin{aligned}
P\left(\bigcup_{t=1}^{\infty} B_t(\delta) \right) &\overset{(a)}{=} P\left(\tau < \infty\right) \\
&\overset{(b)}{=} P\left(\|S_\tau\|^2_{\overline{V}_\tau^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta} \right), \tau < \infty \right) \\
&\overset{(c)}{=} P\left(M_\tau > 1/\delta, \, \tau < \infty\right) \\
&\leq P\left(M_\tau > 1/\delta\right) \\
&\overset{(d)}{\leq} \delta,
\end{aligned}$$

where equalities (a) and (b) follow from the definition of τ, equality (c) follows from Lemma 12, and inequality (d) follows from Markov's inequality. This concludes the proof for Theorem 10. □

We conclude this section by briefly discussing a special case. If for any t, the elements of η_t are statistically independent conditioning on $\mathcal{H}_{t-1}$, then we can prove a variant of Theorem 10: with $\overline{V}_t = V + \sum_{s=1}^{t} X_s X_s^T$ and $S_t = \sum_{s=1}^{t} X_s \eta_s$, Equation A.74 holds with probability at least $1 - \delta$. To see it, notice that in this case

$$\mathbb{E}\left[\exp\left(\lambda^T X_t \eta_t\right) \middle| \mathcal{H}_{t-1}\right] = \mathbb{E}\left[\prod_{k=1}^{K} \exp\left((X_t^T \lambda)(k) \eta_t(k) \right) \middle| \mathcal{H}_{t-1} \right]$$

$$
\begin{aligned}
&\overset{(a)}{=} \prod_{k=1}^{K} \mathbb{E}\left[\exp\left((X_t^T\lambda)(k)\eta_t(k)\right)\middle|\mathcal{H}_{t-1}\right] \\
&\overset{(b)}{\leq} \prod_{k=1}^{K} \exp\left(\frac{(X_t^T\lambda)(k)^2}{2}\right) = \exp\left(\frac{\left\|X_t^T\lambda\right\|^2}{2}\right),
\end{aligned}
\tag{A.77}
$$

where (k) denote the k-th element of the vector. Note that the equality (a) follows from the conditional independence of the elements in η_t, and inequality (b) follows from $|\eta_t(k)| \leq 1$ for all t and k. Thus, if we redefine $D_t^\lambda = \exp\left(\lambda^T X_t \eta_t - \frac{1}{2}\|X_t^T\lambda\|_2^2\right)$, and $M_t^\lambda = \prod_{s=1}^{t} D_s^\lambda$, we can prove that $\{M_t^\lambda\}_t$ is a supermartingale. Consequently, using similar analysis techniques, we can prove the variant of Theorem 10 discussed in this paragraph.

Appendix B

Supplementary for Chapter 3

B.1 Learning the Graph

In the main paper, we assumed that the graph is known, but in practice such a user-user graph may not be available. In such a case, we explore a heuristic to learn the graph on the fly. The computational gains described in the main paper make it possible to simultaneously learn the user-preferences and infer the graph between users in an efficient manner. Our approach for learning the graph is related to methods proposed for multitask and multilabel learning in the batch setting (Gonçalves et al., 2015; Goncalves et al., 2014) and multitask learning in the online setting (Saha et al., 2011). However, prior works that learn the graph in related settings only tackle problem with tens or hundreds of tasks/labels while we learn the graph and preferences across thousands of users.

Let $V_t \in \mathbb{R}^{n \times n}$ be the inverse covariance matrix corresponding to the graph inferred between users at round t. Since zeroes in the inverse covariance matrix correspond to conditional independences between the corresponding nodes (users) (Rue and Held, 2005), we use L1 regularization on V_t for encouraging sparsity in the inferred graph. We use an additional regularization term $\Delta(V_t || V_{t-1})$ to encourage the graph to change smoothly across rounds. This encourages V_t to be close to V_{t-1} according to a distance metric Δ. Following (Saha et al., 2011), we choose Δ to be the log-determinant Bregman divergence given by $\Delta(X||Y) = \mathrm{Tr}(XY^{-1}) - \log |XY^{-1}| - dn$. If $W_t \in R^{d \times n} = [\theta_1 \theta_2 \dots \theta_n]$ corresponds to the matrix of user preference estimates, the combined objective can be written as:

$$[\theta_t, V_t] = \arg\min_{\theta, V} ||\mathbf{r}_t - \Phi_t \theta||_2^2 + \mathrm{Tr}\left(V(\lambda W^T W + V_{t-1}^{-1})\right) + \lambda_2 ||V||_1 - (dn+1)\ln|V| \quad \text{(B.1)}$$

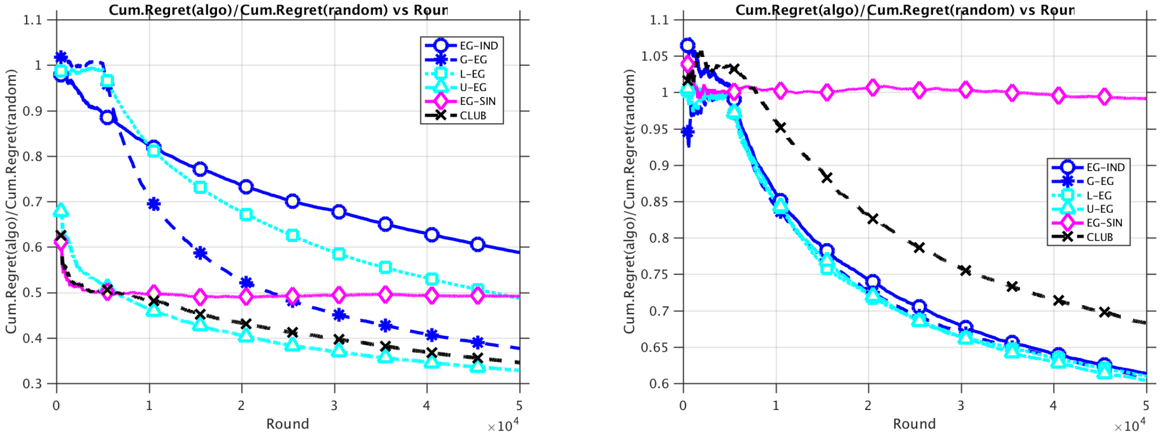
The first term in (B.1) is the data fitting term. The second term imposes the smoothness constraint across the graph and ensures that the changes in V_t are smooth. The third term ensures that the learnt precision matrix is sparse, whereas the last term penalizes the complexity of the precision matrix. This function is independently convex in both θ and V (but not jointly convex), and we alternate between solving for θ_t and V_t in each round. With a fixed V_t, the θ sub-problem is the same as the MAP estimation in the main paper and can be done efficiently. For a fixed θ_t, the V sub-problem is given by

$$V_t = \arg\min_{V} \mathrm{Tr}\left((V[\lambda \overline{W}_t^T \overline{W}_t + V_{t-1}^{-1})\right) + \lambda_2 ||V||_1 - (dn+1)\ln|V| \quad \text{(B.2)}$$

Here $\overline{W}_t$ refers to the mean subtracted (for each dimension) matrix of user preferences. This problem can be written as a graphical lasso problem (Friedman et al., 2008), $\min_X \mathrm{Tr}(SX) + \lambda_2||X||_1 - \log|X|$, where the empirical covariance matrix S is equal to $\lambda \overline{W}_t^T \overline{W}_t + V_{t-1}^{-1}$. We use the highly-scalable second order methods described in (Hsieh et al., 2011, 2013) to solve (B.2). Thus, both sub-problems in the alternating minimization framework at each round can be solved efficiently.

For our preliminary experiments in this direction, we use the most scalable epoch-greedy algorithm for learning the graph on the fly and denote this version as L-EG. We also consider another variant, U-EG in which we start from the Laplacian matrix L corresponding to the given graph and allow it to change by re-estimating the graph according to (B.2). Since U-EG has the flexibility to infer a better graph than the one given, such a variant is important for cases where the prior is meaningful but somewhat misspecified (the given graph accurately reflects some but not all of the user similarities). Similar to (Saha et al., 2011), we start off with an empty graph and start learning the graph only after the preference vectors have become stable, which happens in this case after each user has received 10 recommendations. We update the graph every 1K rounds. For both datasets, we allow the learnt graph to contain at most 100K edges and tune λ_2 to achieve a sparsity level equal to 0.05 in both cases.

To avoid clutter, we plot all the variants of the EG algorithm, L-EG and U-EG, and



(a) Last.fm

(b) Delicious

Figure B.1: Regret Minimization while learning the graph

use EG-IND, G-EG, EG-SIN as baselines. We also plot CLUB as a baseline. For the Last.fm dataset (Figure B.1b(a)), U-EG performs slightly better than G-EG, which already performed well. The regret for L-EG is lower compared to LINUCB-IND indicating that learning the graph helps, but is worse as compared to both CLUB and LINUCB-SIN. On the other hand, for Delicious (Figure B.1b(b)), L-EG and U-EG are the best performing methods. L-EG slightly outperforms EG-IND, underscoring the importance of learning the user-user graph and transferring information between users. It also outperforms G-EG, which implies that it is able to learn a graph which reflects user similarities better than the existing social network between users. For both datasets, U-EG is among the top performing methods, which implies that allowing modifications to a good (in that it reflects user similarities reasonably well) initial graph to model the obtained data might be a good method to overcome prior misspecification. From a scalability point of view, for Delicious the running time for L-EG is 0.1083 seconds/iteration (averaged across T) as compared to 0.04 seconds/iteration for G-EG. This shows that even in the absence of an explicit user-user graph, it is possible to achieve a low regret in an efficient manner.

B.2 Regret bound for Epoch-Greedy

Theorem 11. *Under the additional assumption that $||w_t||_2 \leq 1$ for all rounds t, the expected regret obtained by epoch-greedy in the GOB framework is given as:*

$$R(T) = \widetilde{O}\left(n^{1/3}\left(\frac{\mathrm{Tr}(L^{-1})}{\lambda n}\right)^{\frac{1}{3}} T^{\frac{2}{3}}\right) \tag{B.3}$$

Proof. Let $\mathcal{H}$ be the class of hypotheses of linear functions (one for each user) coupled with Laplacian regularization. Let $\mu(\mathcal{H}, q, s)$ represent the regret or cost of performing s exploitation steps in epoch q. Let the number of exploitation steps in epoch q be s_q.

Lemma 13 (Corollary 3.1 from (Langford and Zhang, 2008))**.** *If $s_q = \lfloor \frac{1}{\mu(\mathcal{H},q,1)} \rfloor$ and Q_T is the minimum Q such that $Q + \sum_{q=1}^{Q} s_q \geq T$, then the regret obtained by Epoch Greedy is bounded by $R(T) \leq 2Q_T$.*

We now bound the quantity $\mu(\mathcal{H}, q, 1)$. Let $Err(q, \mathcal{H})$ be the generalization error for $\mathcal{H}$ after obtaining q unbiased samples in the exploration rounds. Clearly,

$$\mu(\mathcal{H}, q, s) = s \cdot Err(q, \mathcal{H}). \tag{B.4}$$

Let ℓ_{LS} be the least squares loss. Let the number of unbiased samples per user be equal to p. The empirical Rademacher complexity for our hypotheses class $\mathcal{H}$ under ℓ_{LS} can be given as $\widehat{\mathcal{R}}^n_p(\ell_{LS} \circ \mathcal{H})$. The generalization error for $\mathcal{H}$ can be bounded as follows:

Lemma 14 (Theorem 1 from (Maurer, 2006))**.** *With probability $1 - \delta$,*

$$Err(q, \mathcal{H}) \leq \widehat{\mathcal{R}}^n_p(\ell_{LS} \circ \mathcal{H}) + \sqrt{\frac{9\ln(2/\delta)}{2pn}} \tag{B.5}$$

Assume that the target user is chosen uniformly at random. This implies that the expected number of samples per user is at least $p = \lfloor \frac{q}{n} \rfloor$. For simplicity, assume q is exactly divisible by n so that $p = \frac{q}{n}$ (this only affects the bound by a constant factor). Substituting p in (B.5), we obtain

$$Err(q, \mathcal{H}) \leq \widehat{\mathcal{R}}^n_p(\ell_{LS} \circ \mathcal{H}) + \sqrt{\frac{9\ln(2/\delta)}{2q}}. \tag{B.6}$$

The Rademacher complexity can be bounded using Lemma 15 (see below) as follows:

$$\widehat{\mathcal{R}}_p^n(\ell_{LS} \circ \mathcal{H}) \leq \frac{1}{\sqrt{p}}\sqrt{\frac{48\,\mathrm{Tr}(L^{-1})}{\lambda n}} = \frac{1}{\sqrt{q}}\sqrt{\frac{48\,\mathrm{Tr}(L^{-1})}{\lambda}} \tag{B.7}$$

Substituting this into (B.6) we obtain

$$Err(q, \mathcal{H}) \leq \frac{1}{\sqrt{q}}\left[\sqrt{\frac{48\,\mathrm{Tr}(L^{-1})}{\lambda}} + \sqrt{\frac{9\ln(2/\delta)}{2}}\right]. \tag{B.8}$$

We set $s_q = \frac{1}{Err(q,\mathcal{H})}$. Denoting $\left[\sqrt{\frac{48\,\mathrm{Tr}(L^{-1})}{\lambda}} + \sqrt{\frac{9\ln(2/\delta)}{2}}\right]$ as C, $s_q = \frac{\sqrt{q}}{C}$.

Recall that from Lemma 13, we need to determine Q_T such that

$$Q_T + \sum_{q=1}^{Q_T} s_q \geq T \implies \sum_{q=1}^{Q_T}(1 + s_q) \geq T$$

Since $s_q \geq 1$, this implies that $\sum_{q=1}^{Q_T} 2s_q \geq T$. Substituting the value of s_q and observing that for all q, $s_{q+1} \geq s_q$, we obtain the following:

$$2Q_T s_{Q_T} \geq T \implies 2\frac{Q_T^{3/2}}{C} \geq T \implies Q_T \geq \left(\frac{CT}{2}\right)^{\frac{2}{3}}$$

$$Q_T = \left[\sqrt{\frac{12\,\mathrm{Tr}(L^{-1})}{\lambda}} + \sqrt{\frac{9\ln(2/\delta)}{8}}\right]^{\frac{2}{3}} T^{\frac{2}{3}} \tag{B.9}$$

Using the above equation with Lemma 13, we can bound the regret as

$$R(T) \leq 2\left[\sqrt{\frac{12\,\mathrm{Tr}(L^{-1})}{\lambda}} + \sqrt{\frac{9\ln(2/\delta)}{8}}\right]^{\frac{2}{3}} T^{\frac{2}{3}} \tag{B.10}$$

To simplify this expression, we suppress the term $\sqrt{\frac{9\ln(2/\delta)}{8}}$ in the $\widetilde{O}$ notation, implying that

$$R(T) = \widetilde{O}\left(2\left[\frac{12\,\mathrm{Tr}(L^{-1})}{\lambda}\right]^{\frac{1}{3}} T^{\frac{2}{3}}\right) \tag{B.11}$$

To present and interpret the result, we keep only the factors which are dependent on n, λ, L and T. We then obtain

$$R(T) = \widetilde{O}\left(n^{1/3}\left(\frac{\mathrm{Tr}(L^{-1})}{\lambda n}\right)^{\frac{1}{3}} T^{\frac{2}{3}}\right) \tag{B.12}$$

□

This proves Theorem 11. We now prove Lemma 15, which was used to bound the Rademacher complexity.

Lemma 15. *The empirical Rademacher complexity for $\mathcal{H}$ under ℓ_{LS} on observing p unbiased samples for each of the n users can be given as:*

$$\widehat{\mathcal{R}}_p^n(\ell_{LS} \circ \mathcal{H}) \leq \frac{1}{\sqrt{p}}\sqrt{\frac{48\,\mathrm{Tr}(L^{-1})}{\lambda n}} \tag{B.13}$$

Proof.

The Rademacher complexity for a class of linear predictors with graph regularization for a $0/1$ loss function $\ell_{0,1}$ can be bounded using Theorem 2 of (Maurer, 2006). Specifically,

$$\widehat{\mathcal{R}}_p^n(\ell_{0,1} \circ \mathcal{H}) \leq \frac{2M}{\sqrt{p}}\sqrt{\frac{\mathrm{Tr}((\lambda L)^{-1})}{n}} \tag{B.14}$$

where M is the upper bound on the value of $\frac{||L^{1/2}W^*||_2}{\sqrt{n}}$ and W^* is the $d \times n$ matrix corresponding to the true user preferences. We now upper bound $\frac{||L^{\frac{1}{2}}W^*||_2}{\sqrt{n}}$.

$$
\begin{aligned}
||L^{\frac{1}{2}}W^*||_2 &\leq ||L^{\frac{1}{2}}||_2||W^*||_2 \\
||W^*||_2 &\leq ||W^*||_F = \sqrt{\sum_{i=1}^{n} ||w_i^*||_2^2} \\
||W^*||_2 &\leq \sqrt{n} && \text{(Using assumption 1: For all } i,\ ||w_i^*||_2 \leq 1)
\end{aligned}
$$

$$||L^{\frac{1}{2}}|| \leq \nu_{max}(L^{\frac{1}{2}}) = \sqrt{\nu_{max}(L)} \leq \sqrt{3}$$

(The maximum eigenvalue of any normalized Laplacian L_G is 2 (Chung) and recall that $L = L_G + I_n$)

$$\implies \frac{||L^{\frac{1}{2}} W^*||_2}{\sqrt{n}} \leq \sqrt{3} \implies M = \sqrt{3}$$

Since we perform regression using a least squares loss function instead of classification, the Rademacher complexity in our case can be bounded using Theorem 12 from (Bartlett and Mendelson, 2003). Specifically, if ρ is the Lipschitz constant of the least squares problem,

$$\widehat{\mathcal{R}}_p^n(\ell_{LS} \circ \mathcal{H}) \leq 2\rho \cdot \mathcal{R}_p^n(\ell_{0,1} \circ \mathcal{H}) \tag{B.15}$$

Since the estimates $w_{i,t}$ are bounded from above by 1 (additional assumption in the theorem), $\rho = 1$. From Equations B.14, B.15 and the bound on M, we obtain that

$$\widehat{\mathcal{R}}_p^n(\ell_{LS} \circ \mathcal{H}) \leq \frac{4}{\sqrt{p}} \sqrt{\frac{3\,\mathrm{Tr}(L^{-1})}{\lambda n}} \tag{B.16}$$

□

B.3 Regret bound for Thompson Sampling

Theorem 12. *Under the following additional technical assumptions: (a)* $\log(K) < (dn - 1)\ln(2)$ *(b)* $\lambda < dn$ *(c)* $\log\left(\frac{3+T/\lambda dn}{\delta}\right) \leq \log(KT)\log(T/\delta)$*, with probability* $1-\delta$*, the regret obtained by Thompson Sampling in the GOB framework is given as:*

$$R(T) = \widetilde{O}\left(\frac{dn}{\sqrt{\lambda}}\sqrt{T}\sqrt{\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)}\right) \tag{B.17}$$

Proof. We can interpret graph-based TS as being equivalent to solving a single dn-dimensional contextual bandit problem, but with a modified prior covariance ($(L \otimes I_d)^{-1}$ instead of I_{dn}). Our argument closely follows the proof structure in (Agrawal and Goyal, 2012b), but is modified to include the prior covariance. For ease of exposition, assume that the target user at each round is implicit. We use j to index the available items. Let the index of the optimal item at round t be j_t^*, whereas the index of the item chosen by our algorithm is denoted j_t.

Let $\widehat{r}_t(j)$ be the estimated rating of item j at round t. Then, for all j,

$$\widehat{r}_t(j) \sim \mathcal{N}(\langle \theta_t, \boldsymbol{\phi}_j \rangle, s_t(j)) \tag{B.18}$$

Here, $s_t(j)$ is the standard deviation in the estimated rating for item j at round t. Recall that Σ_{t-1} is the covariance matrix at round t. $s_t(j)$ is given as:

$$s_t(j) = \sqrt{\boldsymbol{\phi}_j^T \Sigma_{t-1}^{-1} \boldsymbol{\phi}_j} \tag{B.19}$$

We drop the argument in $s_t(j_t)$ to denote the standard deviation and estimated rating for the selected item j_t i.e. $s_t = s_t(j_t)$ and $\widehat{r}_t = \widehat{r}_t(j_t)$.

Let Δ_t measure the immediate regret at round t incurred by selecting item j_t instead of the optimal item j_t^*. The immediate regret is given by:

$$\Delta_t = \langle \theta^*, \boldsymbol{\phi}_{j_t^*} \rangle - \langle \theta^*, \boldsymbol{\phi}_{j_t} \rangle \tag{B.20}$$

Define $\mathcal{E}^\mu(t)$ as the event such that for all j,

$$\mathcal{E}^\mu(t): \quad |\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| \leq l_t s_t(j) \tag{B.21}$$

Here $l_t = \sqrt{dn \log\left(\frac{3+t/\lambda dn}{\delta}\right)} + \sqrt{3\lambda}$. If the event $\mathcal{E}^\mu(t)$ holds, it implies that the expected rating at round t is close to the true rating with high probability.

Recall that $|\mathcal{C}_t| = K$ and that $\widetilde{\theta}_t$ is a sample drawn from the posterior distribution at round t. Define $\rho_t = \sqrt{9dn \log\left(\frac{t}{\delta}\right)}$ and $g_t = \min\{\sqrt{4dn\ln(t)}, \sqrt{4\log(tK)}\}\rho_t + l_t$. Define $\mathcal{E}^\theta(t)$ as the event such that for all j,

$$\mathcal{E}^\theta(t): \quad |\langle \widetilde{\theta}_t, \boldsymbol{\phi}_j \rangle - \langle \theta_t, \boldsymbol{\phi}_j \rangle| \leq \min\{\sqrt{4dn\ln(t)}, \sqrt{4\log(tK)}\}\rho_t s_t(j) \tag{B.22}$$

If the event $\mathcal{E}^{\theta}(t)$ holds, it implies that the estimated rating using the sample $\widetilde{\theta}_t$ is close to the expected rating at round t.

(B.23)

In lemma 18, we prove that the event $\mathcal{E}^{\mu}(t)$ holds with high probability. Formally, for $\delta \in (0, 1)$,

$$p(\mathcal{E}^{\mu}(t)) \geq 1 - \delta \tag{B.24}$$

To show that the event $\mathcal{E}^{\theta}(t)$ holds with high probability, we use the following lemma from (Agrawal and Goyal, 2012b).

Lemma 16 (Lemma 2 of (Agrawal and Goyal, 2012b))**.**

$$p(\mathcal{E}^{\theta}(t))|\mathcal{F}_{t-1}) \geq 1 - \frac{1}{t^2} \tag{B.25}$$

Next, we use the following lemma to bound the immediate regret at round t.

Lemma 17 (Lemma 4 in (Agrawal and Goyal, 2012b))**.** *Let* $\gamma = \frac{1}{4e\sqrt{\pi}}$*. If the events* $\mathcal{E}^{\mu}(t)$ *and* $\mathcal{E}^{\theta}(t)$ *are true, then for any filtration* $\mathcal{F}_{t-1}$*, the following inequality holds:*

$$\mathbb{E}[\Delta_t|\mathcal{F}_{t-1}] \leq \frac{3g_t}{\gamma}\mathbb{E}[s_t|\mathcal{F}_{t-1}] + \frac{2g_t}{\gamma t^2} \tag{B.26}$$

Define $\mathcal{I}(\mathcal{E})$ to be the indicator function for an event $\mathcal{E}$. Let $regret(t) = \Delta_t \cdot \mathcal{I}(\mathcal{E}^{\mu}(t))$. We use Lemma 19 (proof is given later) which states that with probability at least $1 - \frac{\delta}{2}$,

$$\sum_{t=1}^{T} regret(t) \leq \sum_{t=1}^{T} \frac{3g_t}{\gamma} s_t + \sum_{t=1}^{T} \frac{2g_t}{\gamma t^2} + \sqrt{2 \sum_{t=1}^{T} \frac{36 g_t^2}{\gamma^2} \ln(2/\delta)} \tag{B.27}$$

From Lemma 18, we know that event $\mathcal{E}^{\mu}(t)$ holds for all t with probability at least $1 - \frac{\delta}{2}$. This implies that, with probability $1 - \frac{\delta}{2}$, for all t

$$regret(t) = \Delta_t \tag{B.28}$$

From Equations B.27 and B.28, we have that with probability $1 - \delta$,

$$R(T) = \sum_{t=1}^{T} \Delta_t \leq \sum_{t=1}^{T} \frac{3g_t}{\gamma} s_t + \sum_{t=1}^{T} \frac{2g_t}{\gamma t^2} + \sqrt{2 \sum_{t=1}^{T} \frac{36 g_t^2}{\gamma^2} \ln(2/\delta)}$$

Note that g_t increases with t i.e. for all t, $g_t \leq g_T$

$$R(T) \leq \frac{3g_T}{\gamma} \sum_{t=1}^{T} s_t + \frac{2g_T}{\gamma} \sum_{t=1}^{T} \frac{1}{t^2} + \frac{6g_T}{\gamma} \sqrt{2T \ln(2/\delta)} \tag{B.29}$$

Using Lemma 20 (proof given later), we have the following bound on $\sum_{t=1}^{T} s_t$, the variance of the selected items:

$$\sum_{t=1}^{T} z \leq \sqrt{dnT} \sqrt{C \log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} \tag{B.30}$$

where $C = \frac{1}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)}$.

$$\tag{B.31}$$

Substituting this into Equation B.29, we get

$$R(T) \leq \frac{3g_T}{\gamma} \sqrt{dnT} \sqrt{C \log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} + \frac{2g_T}{\gamma} \sum_{t=1}^{T} \frac{1}{t^2} + \frac{6g_T}{\gamma} \sqrt{2T \ln(2/\delta)}$$

Using the fact that $\sum_{t=1}^{T} \frac{1}{t^2} < \frac{\pi^2}{6}$

$$R(T) \leq \frac{3g_T}{\gamma} \sqrt{dnT} \sqrt{C \log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} + \frac{\pi^2 g_T}{3\gamma} + \frac{6g_T}{\gamma} \sqrt{2T \ln(2/\delta)} \tag{B.32}$$

We now upper bound g_T. By our assumption on K, $\log(K) < (dn-1)\ln(2)$. Hence for all $t \geq 2$, $\min\{\sqrt{4dn\ln(t)}, \sqrt{4\log(tK)}\} = \sqrt{4\log(tK)}$. Hence,

$$\begin{aligned} g_T &= 6\sqrt{dn\log(KT)\log(T/\delta)} + l_T \\ &= 6\sqrt{dn\log(KT)\log(T/\delta)} + \sqrt{dn\log\left(\frac{3+T/\lambda dn}{\delta}\right)} + \sqrt{3\lambda} \end{aligned}$$

By our assumption on λ, $\lambda < dn$. Hence,

$$g_T \leq 8\sqrt{dn\log(KT)\log(T/\delta)} + \sqrt{dn\log\left(\frac{3+T/\lambda dn}{\delta}\right)}$$

Using our assumption that $\log\left(\frac{3+T/\lambda dn}{\delta}\right) \leq \log(KT)\log(T/\delta)$,

$$g_T \leq 9\sqrt{dn\log(KT)\log(T/\delta)} \tag{B.33}$$

Substituting the value of g_T into Equation B.32, we obtain the following:

$$\begin{aligned} R(T) \leq{}& \frac{27dn}{\gamma}\sqrt{T}\sqrt{C\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} \\ &+ \frac{3\pi^2\sqrt{dn\ln(T/\delta)\ln(KT)}}{\gamma} + \frac{54\sqrt{dn\ln(T/\delta)\ln(KT)}\sqrt{2T\ln(2/\delta)}}{\gamma} \end{aligned}$$

For ease of exposition, we keep the just leading terms on d, n and T. This gives the following bound on $R(T)$.

$$R(T) = \widetilde{O}\left(\frac{27dn}{\gamma}\sqrt{T}\sqrt{C\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)}\right)$$

Rewriting the bound to keep only the terms dependent on d, n, λ, T and L. We thus obtain the following equation.

$$R(T) = \widetilde{O}\left(\frac{dn}{\sqrt{\lambda}}\sqrt{T}\sqrt{\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)}\right) \tag{B.34}$$

This proves the theorem. □

We now prove the the auxiliary lemmas used in the above proof.

In the following lemma, we prove that $\mathcal{E}^{\mu}(t)$ holds with high probability, i.e., the expected rating at round t is close to the true rating with high probability.

Lemma 18.

The following statement is true for all $\delta \in (0,1)$:

$$\Pr(\mathbb{E}^{\mu}(t)) \geq 1 - \delta \tag{B.35}$$

Proof.

Recall that $r_t = \langle \theta^*, \phi_{j_t} \rangle + \eta_t$ (Assumption 2) and that $\Sigma_t \theta_t = \frac{b_t}{\sigma^2}$. Define $\mathbf{S}_{t-1} = \sum_{l=1}^{t-1} \eta_l \boldsymbol{\phi}_{j_l}$.

$$\mathbf{S}_{t-1} = \sum_{l=1}^{t-1} \left(r_l - \langle \theta^*, \phi_{j_l} \rangle\right) \boldsymbol{\phi}_{j_l} = \sum_{l=1}^{t-1} \left(r_l \boldsymbol{\phi}_{j_l} - \boldsymbol{\phi}_{j_l} \boldsymbol{\phi}_{j_l}^T \theta^*\right)$$

$$\mathbf{S}_{t-1} = b_{t-1} - \sum_{l=1}^{t-1} \left(\boldsymbol{\phi}_{j_l} \boldsymbol{\phi}_{j_l}^T\right) \theta^* = b_{t-1} - \sigma^2(\Sigma_{t-1} - \Sigma_0)\theta^* = \sigma^2(\Sigma_{t-1}\theta_t - \Sigma_{t-1}\theta^* + \Sigma_0\theta^*)$$

$$\widehat{\theta}_t - \theta^* = \Sigma_{t-1}^{-1}\left(\frac{\mathbf{S}_{t-1}}{\sigma^2} - \Sigma_0 \theta^*\right)$$

The following holds for all j:

$$\begin{aligned} |\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| &= |\langle \boldsymbol{\phi}_j, \theta_t - \theta^* \rangle| \\ &\leq \left| \boldsymbol{\phi}_j^T \Sigma_{t-1}^{-1} \left(\frac{\mathbf{S}_{t-1}}{\sigma^2} - \Sigma_0 \theta^*\right) \right| \end{aligned}$$

$$\leq ||\boldsymbol{\phi}_j||_{\Sigma_{t-1}^{-1}} \left(\left|\left| \frac{\mathbf{S}_{t-1}}{\sigma^2} - \Sigma_0 \theta^* \right|\right|_{\Sigma_{t-1}^{-1}} \right) \quad \text{(Since } \Sigma_{t-1}^{-1} \text{ is positive definite)}$$

By triangle inequality,

$$|\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| \leq ||\boldsymbol{\phi}_j||_{\Sigma_{t-1}^{-1}} \left(\left|\left| \frac{\mathbf{S}_{t-1}}{\sigma^2} \right|\right|_{\Sigma_{t-1}^{-1}} + ||\Sigma_0 \theta^*||_{\Sigma_{t-1}^{-1}} \right) \tag{B.36}$$

We now bound the term $||\Sigma_0 \theta^*||_{\Sigma_{t-1}^{-1}}$

$$\begin{aligned}
||\Sigma_0 \theta^*||_{\Sigma_{t-1}^{-1}} &\leq ||\Sigma_0 \theta^*||_{\Sigma_0^{-1}} = \sqrt{\theta^{*T} \Sigma_0^T \Sigma_0^{-1} \Sigma_0 \theta^*} && \text{(Since } \phi_{j_t} \phi_{j_t}^T \text{ is positive definite for all } t) \\
&= \sqrt{\theta^{*T} \Sigma_0 \theta^*} && \text{(Since } \Sigma_0 \text{ is symmetric)} \\
&\leq \sqrt{\nu_{max}(\Sigma_0)} ||\theta^*||_2 \\
&\leq \sqrt{\nu_{max}(\lambda L \otimes I_d)} && (||\theta^*||_2 \leq 1) \\
&= \sqrt{\nu_{max}(\lambda L)} && (\nu_{max}(A \otimes B) = \nu_{max}(A) \cdot \nu_{max}(B)) \\
&\leq \sqrt{\lambda \cdot \nu_{max}(L)} \\
||\Sigma_0 \theta^*||_{\Sigma_{t-1}^{-1}} &\leq \sqrt{3\lambda}
\end{aligned}$$

(The maximum eigenvalue of any normalized Laplacian is 2 (Chung) and recall that $L = L_G + I_n$)

For bounding $||\boldsymbol{\phi}_j||_{\Sigma_{t-1}^{-1}}$, note that

$$||\boldsymbol{\phi}_j||_{\Sigma_{t-1}^{-1}} = \sqrt{\boldsymbol{\phi}_j^T \Sigma_{t-1}^{-1} \boldsymbol{\phi}_j} = s_t(j)$$

Using the above relations, Equation B.36 can thus be rewritten as:

$$|\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| \leq s_t(j) \left(\frac{1}{\sigma} ||\mathbf{S}_{t-1}||_{\Sigma_{t-1}^{-1}} + \sqrt{3\lambda} \right) \tag{B.37}$$

To bound $||\mathbf{S}_{t-1}||_{\Sigma_{t-1}^{-1}}$, we use Theorem 1 from (Abbasi-Yadkori et al., 2011) which we restate in our context. Note that using this theorem with the prior covariance equal to I_{dn}

gives Lemma 8 of (Agrawal and Goyal, 2012b).

Theorem 13 (Theorem 1 of (Abbasi-Yadkori et al., 2011))**.** *For any* $\delta > 0$*,* $t \geq 1$*, with probability at least* $1 - \delta$*,*

$$||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} \leq 2\sigma^2 \log\left(\frac{\det(\Sigma_t)^{1/2}\det(\Sigma_0)^{-1/2}}{\delta}\right)$$

$$||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} \leq 2\sigma^2\Bigg(\log\left(\det(\Sigma_t)^{1/2}\right) + \log\left(\det(\Sigma_0^{-1})^{1/2}\right) - \log(\delta)\Bigg)$$

Rewriting the above equation,

$$||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} \leq \sigma^2\Bigg(\log\left(\det(\Sigma_t)\right) + \log\left(\det(\Sigma_0^{-1})\right) - 2\log(\delta)\Bigg)$$

We now use the trace-determinant inequality. For any $n \times n$ matrix A, $\det(A) \leq \left(\frac{Tr(A)}{n}\right)^n$ which implies that $\log(\det(A)) \leq n \log\left(\frac{Tr(A)}{n}\right)$. Using this for both Σ_t and Σ_0^{-1}, we obtain:

$$||\mathbf{S}_{t-1}||_{\Sigma^{-1}_{t-1}} \leq dn\sigma^2\Bigg(\log\left(\left(\frac{\text{Tr}(\Sigma_t)}{dn}\right)\right) + \log\left(\left(\frac{\text{Tr}(\Sigma_0^{-1})}{dn}\right)\right) - \frac{2}{dn}\log(\delta)\Bigg) \tag{B.38}$$

Next, we use the fact that

$$\Sigma_t = \Sigma_0 + \sum_{l=1}^{t} \boldsymbol{\phi}_{j_l}\boldsymbol{\phi}_{j_l}^T \implies \text{Tr}(\Sigma_t) \leq \text{Tr}(\Sigma_0) + t \qquad \text{(Since } ||\boldsymbol{\phi}_{j_l}||_2 \leq 1\text{)}$$

Note that $\text{Tr}(A \otimes B) = \text{Tr}(A) \cdot \text{Tr}(B)$. Since $\Sigma_0 = \lambda L \otimes I_d$, it implies that $\text{Tr}(\Sigma_0) = \lambda d \cdot \text{Tr}(L)$. Also note that $\text{Tr}(\Sigma_0^{-1}) = \text{Tr}((\lambda L)^{-1} \otimes I_d) = \frac{d}{\lambda} \text{Tr}(L^{-1})$. Using these relations in Equation B.38,

$$\begin{aligned}
||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} &\leq dn\sigma^2 \Bigg(\log\left(\frac{\lambda d \,\text{Tr}(L) + t}{dn}\right) + \log\left(\frac{\text{Tr}(L^{-1})}{\lambda n}\right) - \frac{2}{dn}\log(\delta) \Bigg) \\
&\leq dn\sigma^2 \Bigg(\log\left(\frac{\text{Tr}(L)\,\text{Tr}(L^{-1})}{n^2} + \frac{t\,\text{Tr}(L^{-1})}{\lambda dn^2}\right) - \log(\delta^{\frac{2}{dn}}) \Bigg) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\log(a) + \log(b) = \log(ab)) \\
&= dn\sigma^2 \log\left(\frac{\text{Tr}(L)\,\text{Tr}(L^{-1})}{n^2\delta} + \frac{t\,\text{Tr}(L^{-1})}{\lambda dn^2\delta}\right) \qquad (\text{Redefining } \delta \text{ as } \delta^{\frac{2}{dn}})
\end{aligned}$$

If $L = I_n$, $\text{Tr}(L) = \text{Tr}(L^{-1}) = n$, we recover the bound in (Agrawal and Goyal, 2012b) i.e.

$$||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} \leq dn\sigma^2 \log\left(\frac{1 + t/\lambda dn}{\delta}\right) \tag{B.39}$$

The upper bound for $\text{Tr}(L)$ is $3n$, whereas the upper bound on $\text{Tr}(L^{-1})$ is n. We thus obtain the following relation.

$$\begin{aligned}
||\mathbf{S}_{t-1}||^2_{\Sigma^{-1}_{t-1}} &\leq dn\sigma^2 \log\left(\frac{3}{\delta} + \frac{t}{\lambda dn\delta}\right) \\
||\mathbf{S}_{t-1}||_{\Sigma^{-1}_{t-1}} &\leq \sigma\sqrt{dn \log\left(\frac{3 + t/\lambda dn}{\delta}\right)}
\end{aligned} \tag{B.40}$$

Combining Equations B.37 and B.40, we have with probability $1 - \delta$,

$$\begin{aligned}
|\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| &\leq s_t(k) \left(\sqrt{dn \log\left(\frac{3 + t/\lambda dn}{\delta}\right)} + \sqrt{3\lambda} \right) \\
|\langle \theta_t, \boldsymbol{\phi}_j \rangle - \langle \theta^*, \boldsymbol{\phi}_j \rangle| &\leq s_t(k) l_t
\end{aligned}$$

where $l_t = \sqrt{dn \log\left(\frac{3+t/\lambda dn}{\delta}\right)} + \sqrt{3\lambda}$. This completes the proof.

$$\tag{B.41}$$

□

Lemma 19. *With probability* $1-\delta$,

$$\sum_{t=1}^{T} regret(t) \leq \sum_{t=1}^{T} \frac{3g_t}{\gamma} s_t + \sum_{t=1}^{T} \frac{2g_t}{\gamma t^2} + \sqrt{2\sum_{t=1}^{T} \frac{36g_t^2}{\gamma^2} \ln\frac{2}{\delta}} \tag{B.42}$$

Proof.

Let Z_l and Y_t be defined as follows:

$$\begin{aligned}
&Z_l = regret(l) - \frac{3g_l}{\gamma} s_l - \frac{2g_l}{\gamma l^2} \\
&Y_t = \sum_{l=1}^{t} Z_l \qquad (B.43)\\
&\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] = \mathbb{E}[X_t] = \mathbb{E}[regret(t)|\mathcal{F}_{t-1}] - \frac{3g_t}{\gamma} s_t - \frac{2g_t}{\gamma t^2} \\
&\mathbb{E}[regret(t)|\mathcal{F}_{t-1}] \leq \mathbb{E}[\Delta_t|\mathcal{F}_{t-1}] \leq \frac{3g_t}{\gamma} s_t - \frac{2g_t}{\gamma t^2} \\
&\qquad\qquad \text{(Definition of } regret(t) \text{ and using lemma 17)} \\
&\mathbb{E}[Y_t - Y_{t-1}|\mathcal{F}_{t-1}] \leq 0
\end{aligned}$$

Hence, Y_t is a super-martingale process. We now state and use the Azuma-Hoeffding inequality for Y_t

(B.44)

Definition 3 (Azuma-Hoeffding)**.** *If a super-martingale* Y_t *(with* $t \geq 0$*) and its the corresponding filtration* $\mathcal{F}_{t-1}$, *satisfies* $|Y_t - Y_{t-1}| \leq c_t$ *for some constant* c_t, *for all* $t = 1, \ldots T$, *then for any* $a \geq 0$,

$$Pr(Y_T - Y_0 \geq a) \leq exp\left(\frac{-a^2}{2\sum_{t=1}^{T} c_t^2}\right) \tag{B.45}$$

We define $Y_0 = 0$. Note that $|Y_t - Y_{t-1}| = |Z_l|$ is bounded by $1 + \frac{3g_l}{\gamma} - \frac{2g_l}{\gamma l^2}$. Hence, $c_t = \frac{6g_t}{\gamma}$. Setting $a = \sqrt{2\ln(2/\delta)\sum_{t=1}^{T} c_t^2}$ in the above inequality, we obtain that with probability $1 - \frac{\delta}{2}$,

$$Y_T \leq \sqrt{2\sum_{t=1}^{T}\frac{36g_t^2}{\gamma^2}\ln(2/\delta)}$$

$$\sum_{t=1}^{T}\left(regret(t) - \frac{3g_t}{\gamma}s_t - \frac{2g_t}{\gamma t^2}\right) \leq \sqrt{2\sum_{t=1}^{T}\frac{36g_t^2}{\gamma^2}\ln(2/\delta)} \tag{B.46}$$

$$\sum_{t=1}^{T} regret(t) \leq \sum_{t=1}^{T}\frac{3g_t}{\gamma}s_t + \sum_{t=1}^{T}\frac{2g_t}{\gamma t^2} + \sqrt{2\sum_{t=1}^{T}\frac{36g_t^2}{\gamma^2}\ln(2/\delta)} \tag{B.47}$$

□

Lemma 20.

$$\sum_{t=1}^{T} z \leq \sqrt{dnT}\sqrt{C\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} \tag{B.48}$$

Proof.

Following the proof in (Dani et al., 2008; Wen et al., 2015b),

$$\begin{aligned}
\det\left[\Sigma_t\right] &\geq \det\left[\Sigma_{t-1} + \frac{1}{\sigma^2}\boldsymbol{\phi}_{j_t}\boldsymbol{\phi}_{j_t}^T\right] \\
&= \det\left[\Sigma_{t-1}^{\frac{1}{2}}\left(I + \frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}\boldsymbol{\phi}_{j_t}\boldsymbol{\phi}_{j_t}^T\Sigma_{t-1}^{-\frac{1}{2}}\right)\Sigma_{t-1}^{\frac{1}{2}}\right] \\
&= \det\left[\Sigma_{t-1}\right]\det\left[I + \frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}\boldsymbol{\phi}_{j_t}\boldsymbol{\phi}_{j_t}^T\Sigma_{t-1}^{-\frac{1}{2}}\right] \\
\det\left[\Sigma_t\right] &= \det\left[\Sigma_{t-1}\right]\left(1 + \frac{1}{\sigma^2}\boldsymbol{\phi}_{j_t}^T\Sigma_{t-1}^{-1}\boldsymbol{\phi}_{j_t}\right) = \det\left[\Sigma_{t-1}\right]\left(1 + \frac{s_t^2}{\sigma^2}\right) \\
\log\left(\det\left[\Sigma_t\right]\right) &\geq \log\left(\det\left[\Sigma_{t-1}\right]\right) + \log\left(1 + \frac{s_t^2}{\sigma^2}\right)
\end{aligned}$$

$$\log\left(\det\left[\Sigma_T\right]\right) \geq \log\left(\det\left[\Sigma_0\right]\right) + \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) \tag{B.49}$$

If A is an $n \times n$ matrix, and B is an $d \times d$ matrix, then $\det[A \otimes B] = \det[A]^d \det[B]^n$. Hence,

$$\begin{aligned}
&\det[\Sigma_0] = \det[\lambda L \otimes I_d] = \det[\lambda L]^d \\
&\det[\Sigma_0] = [\lambda^n \det(L)]^d = \lambda^{dn}[\det(L)]^d \\
&\log\left(\det[\Sigma_0]\right) = dn \log\left(\lambda\right) + d \log\left(\det[L]\right) && \text{(B.50)}
\end{aligned}$$

From Equations B.49 and B.50,

$$\log\left(\det\left[\Sigma_T\right]\right) \geq \left(dn \log\left(\lambda\right) + d \log\left(\det[L]\right)\right) + \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) \tag{B.51}$$

We now bound the trace of $\mathrm{Tr}(\Sigma_{T+1})$.

$$\begin{aligned}
&\mathrm{Tr}(\Sigma_{t+1}) = \mathrm{Tr}(\Sigma_t) + \frac{\mathbf{1}}{\sigma^2}\boldsymbol{\phi}_{j_t}\boldsymbol{\phi}_{j_t}^T \implies \mathrm{Tr}(\Sigma_{t+1}) \leq \mathrm{Tr}(\Sigma_t) + \frac{\mathbf{1}}{\sigma^2} && (\text{Since } ||\boldsymbol{\phi}_{j_t}|| \leq 1) \\
&\mathrm{Tr}(\Sigma_T) \leq \mathrm{Tr}(\Sigma_0) + \frac{\mathbf{T}}{\sigma^2}
\end{aligned}$$

Since $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A) \cdot \mathrm{Tr}(B)$

$$\mathrm{Tr}(\Sigma_T) \leq \mathrm{Tr}\left(\lambda(L \otimes I_d)\right) + \frac{T}{\sigma^2} \implies \mathrm{Tr}(\Sigma_T) \leq \lambda d\, \mathrm{Tr}(L) + \frac{T}{\sigma^2} \tag{B.52}$$

Using the determinant-trace inequality, we have the following relation:

$$\begin{aligned}
&\left(\frac{1}{dn}\mathrm{Tr}(\Sigma_T)\right)^{dn} \geq \left(\det[\Sigma_T]\right) \\
&dn \log\left(\frac{1}{dn}\mathrm{Tr}(\Sigma_T)\right) \geq \log\left(\det[\Sigma_T]\right) && \text{(B.53)}
\end{aligned}$$

Using Equations B.51, B.52 and B.53, we obtain the following relation.

$$dn \log\left(\frac{\lambda d \operatorname{Tr}(L) + \frac{T}{\sigma^2}}{dn}\right) \geq \left(dn \log(\lambda) + d \log(\det[L])\right) + \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right)$$

$$\begin{aligned}
\sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) &\leq dn \log\left(\frac{\lambda d \operatorname{Tr}(L) + \frac{T}{\sigma^2}}{dn}\right) - dn \log(\lambda) - d \log(\det[L]) \\
&\leq dn \log\left(\frac{\lambda d \operatorname{Tr}(L) + \frac{T}{\sigma^2}}{dn}\right) - dn \log(\lambda) + d \log\left(\det[L^{-1}]\right) \\
&\qquad\qquad (det[L^{-1}] = 1/det[L]) \\
&\leq dn \log\left(\frac{\lambda d \operatorname{Tr}(L) + \frac{T}{\sigma^2}}{dn}\right) - dn \log(\lambda) + dn \log\left(\frac{1}{n} \operatorname{Tr}(L^{-1})\right) \\
&\qquad\qquad (\text{Using the determinant-trace inequality for } \log(\det[L^{-1}])) \\
&\leq dn \log\left(\frac{\lambda d \operatorname{Tr}(L) \operatorname{Tr}(L^{-1}) + \frac{\operatorname{Tr}(L^{-1})T}{\sigma^2}}{\lambda d n^2}\right) \qquad (\log(a) + \log(b) = \log(ab)) \\
&\leq dn \log\left(\frac{\operatorname{Tr}(L) \operatorname{Tr}(L^{-1})}{n^2} + \frac{\operatorname{Tr}(L^{-1})T}{\lambda d n^2 \sigma^2}\right)
\end{aligned}$$

The maximum eigenvalue of any Laplacian is 2. Hence $\operatorname{Tr}(L)$ is upper-bounded by $3n$.

$$\sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) \leq dn \log\left(\frac{3 \operatorname{Tr}(L^{-1})}{n} + \frac{\operatorname{Tr}(L^{-1})T}{\lambda d n^2 \sigma^2}\right) \tag{B.54}$$

$$\tag{B.55}$$

$$\begin{aligned}
s_t^2 = \boldsymbol{\phi}_j^T \Sigma_t^{-1} \boldsymbol{\phi}_j &\leq \boldsymbol{\phi}_j^T \Sigma_0^{-1} \boldsymbol{\phi}_j \\
&\qquad\qquad (\text{Since we are making positive definite updates at each round } t) \\
&\leq \|\boldsymbol{\phi}_j\|^2 \nu_{max}(\Sigma_0^{-1}) \\
&= \|\boldsymbol{\phi}_j\|^2 \frac{1}{\nu_{min}(\lambda L \otimes I_d)} \\
&= \|\boldsymbol{\phi}_j\|^2 \frac{1}{\nu_{min}(\lambda L)} \qquad (\nu_{min}(A \otimes B) = \nu_{min}(A)\nu_{min}(B))
\end{aligned}$$

$$\leq \frac{1}{\lambda} \cdot \frac{1}{\nu_{min}(L)} \qquad (||\phi_j||_2 \leq 1)$$

$$s_t^2 \leq \frac{1}{\lambda} \qquad \text{(Minimum eigenvalue of a normalized Laplacian } L_G \text{ is 0. } L = L_G + I_n\text{)}$$

Moreover, for all $y \in [0, 1/\lambda]$, we have $\log\left(1 + \frac{y}{\sigma^2}\right) \geq \lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right) y$ based on the concavity of $\log(\cdot)$. To see this, consider the following function:

$$h(y) = \frac{\log\left(1 + \frac{y}{\sigma^2}\right)}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)} - y \tag{B.56}$$

Clearly, $h(y)$ is concave. Also note that, $h(0) = h(1/\lambda) = 0$. Hence for all $y \in [0, 1/\lambda]$, the function $h(y) \geq 0$. This implies that $\log\left(1 + \frac{y}{\sigma^2}\right) \geq \lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right) y$. We use this result by setting $y = s_t^2$.

$$\log\left(1 + \frac{s_t^2}{\sigma^2}\right) \geq \lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right) s_t^2$$

$$s_t^2 \leq \frac{1}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) \tag{B.57}$$

Hence,

$$\sum_{t=1}^{T} s_t^2 \leq \frac{1}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)} \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right) \tag{B.58}$$

By Cauchy Schwartz,

$$\sum_{t=1}^{T} z \leq \sqrt{T} \sqrt{\sum_{t=1}^{T} s_t^2} \tag{B.59}$$

From Equations B.58 and B.59,

$$\sum_{t=1}^{T} z \leq \sqrt{T} \sqrt{\frac{1}{\lambda \log\left(1 + \frac{1}{\lambda\sigma^2}\right)} \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right)}$$

$$\sum_{t=1}^{T} z \leq \sqrt{T} \sqrt{C \sum_{t=1}^{T} \log\left(1 + \frac{s_t^2}{\sigma^2}\right)} \tag{B.60}$$

where $C = \frac{1}{\lambda \log\left(1+\frac{1}{\lambda\sigma^2}\right)}$. Using Equations B.54 and B.60,

$$\begin{aligned}
\sum_{t=1}^{T} z &\leq \sqrt{dnT}\sqrt{C \log\left(\frac{3\,\mathrm{Tr}(L^{-1})}{n} + \frac{\mathrm{Tr}(L^{-1})T}{\lambda d n^2 \sigma^2}\right)} \\
\sum_{t=1}^{T} z &\leq \sqrt{dnT}\sqrt{C \log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda d n \sigma^2}\right)} \qquad \text{(B.61)}
\end{aligned}$$

□

Appendix C

Supplementary for Chapter 4

C.1 Proof for Theorem 6

We prove Theorem 6 in this section. First, we have the following tail bound for Binomial random variables:

Proposition 2 (Binomial Tail Bound)**.** *Assume that random variable $X \sim \mathrm{Bino}(n, p)$, then for any k s.t. $np < k < n$, we have*

$$\mathcal{P}(X \geq k) \leq \exp\left(-nD\left(\frac{k}{n}\middle\| p\right)\right),$$

where $D\left(\frac{k}{n}\middle\|p\right)$ is the KL-divergence between $\frac{k}{n}$ and p.

Please refer to (Arratia and Gordon, 1989) for the proof of Proposition 2.

Notice that for our considered case, the "observation history" of the agent at the beginning of time t is completely characterized by a triple $\mathcal{H}_t = (\alpha_{t-1}, \beta_{t-1}, t)$, where α_{t-1} is the number of times arm 1 has been pulled from time 1 to $t-1$ and the realized reward is 1, plus the pseudo count α_0; similarly, β_{t-1} is the number of times arm 1 has been pulled from time 1 to $t-1$ and the realized reward is 0, plus the pseudo count β_0. Moreover, conditioning on this history $\mathcal{H}_t$, the probability that the agent will pull arm 1 under the NPB only depends on $(\alpha_{t-1}, \beta_{t-1})$. To simplify the exposition, we use $P(\alpha_{t-1}, \beta_{t-1})$ to denote this conditional probability. The following lemma bounds this probability in a "bad" history:

Lemma 21. *Consider a "bad" history $\mathcal{H}_t$ with $\alpha_{t-1} = 1$ and $\beta_{t-1} = 1+m$ for some integer*

$m \geq 15$, *then we have*

$$P(\alpha_{t-1}, \beta_{t-1}) < \exp\left(-(m+2)\log(m+2)/20\right) < \exp\left(-m\log(m)/20\right).$$

Proof. Recall that by definition, we have

$$\begin{aligned}
P(\alpha_{t-1}, \beta_{t-1}) &= P(1, m+1) \\
&\overset{(a)}{=} \mathcal{P}\left(w_t \geq 1/4 \,|\, \alpha_{t-1} = 1, \beta_{t-1} = m+1\right) \\
&= \mathcal{P}\left((m+2)w_t \geq (m+2)/4 \,|\, \alpha_{t-1} = 1, \beta_{t-1} = m+1\right) \\
&\overset{(b)}{\leq} \exp\left(-(m+2)D\left(\frac{1}{4}\middle\|\frac{1}{m+2}\right)\right), \qquad \text{(C.1)}
\end{aligned}$$

where (a) follows from the NPB procedure in this case, and (b) follows from Proposition 2. Specifically, recall that $(m+2)w_t \sim \text{Bino}(m+2, 1/(m+2))$, and $(m+2)/4 > (m+2)\frac{1}{m+2} = 1$ for $m \geq 15$. Thus, the conditions of Proposition 2 hold in this case. Furthermore, we have

$$\begin{aligned}
D\left(\frac{1}{4}\middle\|\frac{1}{m+2}\right) &= \frac{1}{4}\log\left(\frac{m+2}{4}\right) + \frac{3}{4}\log\left(\frac{3(m+2)}{4(m+1)}\right) \\
&\geq \frac{1}{4}\log(m+2) - \frac{1}{4}\log(4) + \frac{3}{4}\log\left(\frac{3}{4}\right) \\
&= \frac{1}{20}\log(m+2) + \left[\frac{1}{5}\log(m+2) - \frac{1}{4}\log(4) + \frac{3}{4}\log\left(\frac{3}{4}\right)\right] \\
&\overset{(c)}{>} \frac{1}{20}\log(m+2), \qquad \text{(C.2)}
\end{aligned}$$

where (c) follows from the fact that $\frac{1}{5}\log(m+2) - \frac{1}{4}\log(4) + \frac{3}{4}\log\left(\frac{3}{4}\right) \geq 0$ for $m > 15$. Thus we have

$$\begin{aligned}
P(\alpha_{t-1}, \beta_{t-1}) &\leq \exp\left(-(m+2)D\left(\frac{1}{4}\middle\|\frac{1}{m+2}\right)\right) \\
&< \exp\left(-(m+2)\log(m+2)/20\right) \\
&< \exp\left(-m\log(m)/20\right). \qquad \text{(C.3)}
\end{aligned}$$

□

The following technical lemma derives the expected value of a truncated geometric random variable, as well as a lower bound on it, which will be used in the subsequent analysis:

Lemma 22 (Expected Value of Truncated Geometric R.V.)**.** *Assume that Z is a truncated geometric r.v. with parameter $p \in (0,1)$ and integer $l \geq 1$. Specifically, the domain of Z is $\{0, 1, \ldots, l\}$, and $\mathcal{P}(Z = i) = (1-p)^i p$ for $i = 0, 1, \ldots, l-1$ and $\mathcal{P}(Z = l) = (1-p)^l$. Then we have*

$$\mathbb{E}(Z) = \left[\frac{1}{p} - 1\right]\left[1 - (1-p)^l\right] \geq \frac{1}{2}\min\left\{\frac{1}{p} - 1,\, l(1-p)\right\}.$$

Proof. Notice that by definition, we have

$$\mathbb{E}(Z) = p \underbrace{\left[\sum_{i=0}^{l-1} i(1-p)^i\right]}_{A} + l(1-p)^l$$

Define the shorthand notation $A = \sum_{i=0}^{l-1} i(1-p)^i$, we have

$$\begin{aligned}(1-p)A &= \sum_{i=0}^{l-1} i(1-p)^{i+1} = \sum_{i=1}^{l}(i-1)(1-p)^i \\ &= \sum_{i=0}^{l} i(1-p)^i - \left[\frac{1}{p} - 1\right]\left[1 - (1-p)^l\right] \\ &= A + l(1-p)^l - \left[\frac{1}{p} - 1\right]\left[1 - (1-p)^l\right]. \end{aligned} \tag{C.4}$$

Recall that $\mathbb{E}(Z) = pA + l(1-p)^l$, we have proved that $\mathbb{E}(Z) = \left[\frac{1}{p} - 1\right]\left[1 - (1-p)^l\right]$.

Now we prove the lower bound. First, we prove that

$$(1-p)^l \leq 1 - \frac{pl}{1+pl} \tag{C.5}$$

always holds by induction on l. Notice that when $l = 1$, the LHS of equation (C.5) is $1 - p$, and the RHS of equation (C.5) is $\frac{1}{1+p}$. Hence, this inequality trivially holds in the base case. Now assume that equation (C.5) holds for l, we prove that it also holds for $l + 1$.

Notice that

$$\begin{aligned}(1-p)^{l+1} &= (1-p)^l(1-p) \overset{(a)}{\leq} \left(1-\frac{pl}{1+pl}\right)(1-p) \\ &= 1-\frac{p(l+1)}{1+pl}+\frac{p}{1+pl}-p+\frac{p^2l}{1+pl} \\ &= 1-\frac{p(l+1)}{1+pl} \leq 1-\frac{p(l+1)}{1+p(l+1)}\,, \end{aligned} \tag{C.6}$$

where (a) follows from the induction hypothesis. Thus equation (C.5) holds for all p and l. Notice that equation C.5 implies that

$$\mathbb{E}(Z) = \left[\frac{1}{p}-1\right]\left[1-(1-p)^l\right] \geq \left[\frac{1}{p}-1\right]\frac{pl}{1+pl}\,.$$

We now prove the lower bound. Notice that for any l, $\frac{pl}{1+pl}$ is an increasing function of p, thus for $p \geq 1/l$, we have

$$\left[\frac{1}{p}-1\right]\frac{pl}{1+pl} \geq \left[\frac{1}{p}-1\right]/2 \geq \frac{1}{2}\min\left\{\frac{1}{p}-1,\, l(1-p)\right\}.$$

On the other hand, if $p \leq 1/l$, we have

$$\left[\frac{1}{p}-1\right]\frac{pl}{1+pl} = \frac{(1-p)l}{1+pl} \geq (1-p)l/2 \geq \frac{1}{2}\min\left\{\frac{1}{p}-1,\, l(1-p)\right\}\,.$$

Combining the above results, we have proved the lower bound on $\mathbb{E}(Z)$. □

We then prove the following lemma:

Lemma 23 (Regret Bound Based on m)**.** *When NPB is applied in the considered case, for any integer m and time horizon T satisfying* $15 \leq m \leq T$*, we have*

$$\mathbb{E}\left[R(T)\right] \geq \frac{2^{-m}}{8}\min\left\{\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{4}\right\}\,.$$

Proof. We start by defining the bad event $\mathcal{G}$ as

$$\mathcal{G} = \left\{\exists t = 1, 2, \ldots \text{ s.t. } \alpha_{t-1} = 1 \text{ and } \beta_{t-1} = m+1\right\}\,.$$

Thus, we have $\mathbb{E}[R(T)] \geq \mathcal{P}(\mathcal{G})\, \mathbb{E}[R(T)|\, \mathcal{G}]$. Since $\alpha_t \geq 1$ for all $t = 1, 2, \ldots$, with probability 1, the agent will pull arm 1 infinitely often. Moreover, the event $\mathcal{G}$ only depends on the outcomes of the first m pulls of arm 1. Thus we have $\mathcal{P}(\mathcal{G}) = 2^{-m}$. Furthermore, conditioning on $\mathcal{G}$, we define the stopping time τ as

$$\tau = \min\{t = 1, 2, \ldots \text{ s.t. } \alpha_{t-1} = 1 \text{ and } \beta_{t-1} = m + 1\} \ .$$

Then we have

$$\begin{aligned}
\mathbb{E}[R(T)] \geq\, & \mathcal{P}(\mathcal{G})\, \mathbb{E}[R(T)|\, \mathcal{G}] = 2^{-m}\mathbb{E}[R(T)|\, \mathcal{G}] \\
=\, & 2^{-m}\left[\mathcal{P}(\tau > T/2|\mathcal{G})\, \mathbb{E}[R(T)|\, \mathcal{G}, \tau > T/2] + \mathcal{P}(\tau \leq T/2|\mathcal{G})\, \mathbb{E}[R(T)|\, \mathcal{G}, \tau \leq T/2]\right] \\
\geq\, & 2^{-m} \min\{\mathbb{E}[R(T)|\, \mathcal{G}, \tau > T/2], \mathbb{E}[R(T)|\, \mathcal{G}, \tau \leq T/2]\} \qquad \text{(C.7)}
\end{aligned}$$

Notice that conditioning on event $\{\mathcal{G}, \tau > T/2\}$, in the first $\lfloor T/2 \rfloor$ steps, the agent either pulls arm 2 or pulls arm 1 but receives a reward 0, thus, by definition of $R(T)$, we have

$$\mathbb{E}[R(T)|\, \mathcal{G}, \tau > T/2] \geq \frac{\lfloor T/2 \rfloor}{4}.$$

On the other hand, if $\tau \leq T/2$, notice that for any time $t \geq \tau$ with history $\mathcal{H}_t = (\alpha_{t-1}, \beta_{t-1}, t)$ s.t. $(\alpha_{t-1}, \beta_{t-1}) = (1, m+1)$, the agent will pull arm 1 conditionally independently with probability $P(1, m+1)$. Thus, conditioning on $\mathcal{H}_\tau$, the number of times the agent will pull arm 2 before it pulls arm 1 again follows the truncated geometric distribution with parameter $P(1, m+1)$ and $T - \tau + 1$. From Lemma 22, for any $\tau \leq T/2$, we have

$$\begin{aligned}
\mathbb{E}[R(T)|\, \mathcal{G}, \tau] &\overset{(a)}{\geq} \frac{1}{8} \min\left\{\frac{1}{P(1, m+1)} - 1,\ (T - \tau + 1)(1 - P(1, m+1))\right\} \\
&\overset{(b)}{\geq} \frac{1}{8} \min\left\{\frac{1}{P(1, m+1)} - 1,\ \frac{T}{2}(1 - P(1, m+1))\right\} \\
&\overset{(c)}{>} \frac{1}{8} \min\left\{\exp\left((m+2)\log(m+2)/20\right) - 1,\ \frac{T}{4}\right\} \\
&\overset{(d)}{\geq} \frac{1}{8} \min\left\{\exp\left(m\log(m)/20\right),\ \frac{T}{4}\right\}, \qquad \text{(C.8)}
\end{aligned}$$

notice that a factor of 1/4 in inequality (a) is due to the reward gap. Inequality (b) follows from the fact that $\tau \le T/2$; inequality (c) follows from Lemma 21, which states that for $m \ge 15$, we have $P(\alpha_{t-1}, \beta_{t-1}) < \exp\left(-(m+2)\log(m+2)/20\right) < \frac{1}{2}$; inequality (d) follows from the fact that for $m \ge 15$, we have

$$\exp\left((m+2)\log(m+2)/20\right) - 1 > \exp\left(m\log(m)/20\right) \,.$$

Finally, notice that

$$\mathbb{E}\left[R(T)|\,\mathcal{G}, \tau \le T/2\right] = \sum_{\tau \le T/2} \mathcal{P}(\tau|\mathcal{G}, \tau \le T/2)\mathbb{E}\left[R(T)|\,\mathcal{G}, \tau\right] > \frac{1}{8}\min\left\{\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{4}\right\} \,.$$

Thus, combining everything together, we have

$$\begin{aligned}\mathbb{E}[R(T)] &\ge 2^{-m}\min\left\{\mathbb{E}\left[R(T)|\,\mathcal{G}, \tau > T/2\right], \mathbb{E}\left[R(T)|\,\mathcal{G}, \tau \le T/2\right]\right\} \\ &> \frac{2^{-m}}{4}\min\left\{\frac{1}{2}\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{8}, \lfloor\frac{T}{2}\rfloor\right\} \\ &= \frac{2^{-m}}{4}\min\left\{\frac{1}{2}\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{8}\right\} \,, \qquad \text{(C.9)}\end{aligned}$$

where the last equality follows from the fact that $\frac{T}{8} < \lfloor\frac{T}{2}\rfloor$ for $T \ge 15$. This concludes the proof. □

Finally, we prove Theorem 6.

Proof. For any given $\gamma \in (0,1)$, we choose $m = \left\lceil \frac{\gamma\log(T)}{2} \right\rceil$. Since

$$T \ge \exp\left[\frac{2}{\gamma}\exp\left(\frac{80}{\gamma}\right)\right] \,,$$

we have

$$T \gg m = \left\lceil \frac{\gamma\log(T)}{2} \right\rceil \ge \exp\left(\frac{80}{\gamma}\right) \ge \exp(80) \gg 15,$$

thus, Lemma 23 is applicable. Notice that

$$\mathbb{E}\left[R(T)\right] \ge \frac{2^{-m}}{8}\min\left\{\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{4}\right\} > \frac{\exp(-m)}{8}\min\left\{\exp\left(\frac{m\log(m)}{20}\right), \frac{T}{4}\right\} \,.$$

Furthermore, we have

$$\exp(-m)T > \exp\left(-\gamma \log(T)\right) T = T^{1-\gamma}\,,$$

where the first inequality follows from $m = \left\lceil \frac{\gamma \log(T)}{2} \right\rceil < \gamma \log(T)$. On the other hand, we have

$$\exp(-m) \exp\left(\frac{m \log(m)}{20}\right) = \exp\left(\frac{m \log(m)}{20} - m\right) \geq \exp\left(\frac{m \log(m)}{40}\right)\,,$$

where the last inequality follows from the fact that $\frac{m \log(m)}{40} \geq m$, since $m \geq \exp(80)$. Notice that we have

$$\exp\left(\frac{m \log(m)}{40}\right) \geq \exp\left(\frac{\gamma \log(T) \log(\frac{\gamma \log(T)}{2})}{80}\right) \geq T\,,$$

where the first inequality follows from the fact that $m \geq \frac{\gamma \log(T)}{2}$, and the second inequality follows from $T \geq \exp\left[\frac{2}{\gamma} \exp\left(\frac{80}{\gamma}\right)\right]$. Putting it together, we have

$$\mathbb{E}\left[R(T)\right] > \frac{1}{8} \min\left\{T, \frac{T^{1-\gamma}}{4}\right\} = \frac{T^{1-\gamma}}{32}\,.$$

This concludes the proof for Theorem 6. □

C.2 Proof for Theorem 2

For simplicity of exposition, we consider 2 arms with means $\mu_1 > \mu_2$. Let $\Delta = \mu_1 - \mu_2$. Let $\overline{\mu}_t(k)$ be the mean of the history of arm k at time t and $\widehat{\mu}_t(k)$ be the mean of the bootstrap sample of arm k at time t. Note that both are random variables. Each arm is initially explored m times. Since $\overline{\mu}_t(k)$ and $\widehat{\mu}_t(k)$ are estimated from random samples of size at least m, we get from Hoeffding's inequality (Theorem 2.8 in Boucheron *et al* (Boucheron et al., 2013)) that

$$\begin{aligned} P(\overline{\mu}_t(1) \leq \mu_1 - \varepsilon) &\leq \exp[-2\varepsilon^2 m]\,, \\ P(\overline{\mu}_t(2) \geq \mu_2 + \varepsilon) &\leq \exp[-2\varepsilon^2 m]\,, \\ P(\widehat{\mu}_t(1) \leq \overline{\mu}_t(1) - \varepsilon) &\leq \exp[-2\varepsilon^2 m]\,, \end{aligned}$$

$$P(\widehat{\mu}_t(2) \geq \overline{\mu}_t(2) + \varepsilon) \leq \exp[-2\varepsilon^2 m]$$

for any $\varepsilon > 0$ and time $t > 2m$. The first two inequalities hold for any μ_1 and μ_2. The last two hold for any $\overline{\mu}_t(1)$ and $\overline{\mu}_t(2)$, and therefore also in expectation over their random realizations. Let $\mathcal{E}$ be the event that the above inequalities hold jointly at all times $t > 2m$ and $\overline{\mathcal{E}}$ be the complement of event $\mathcal{E}$. Then by the union bound,

$$P(\overline{\mathcal{E}}) \leq 4T \exp[-2\varepsilon^2 m] \,.$$

By the design of the algorithm, the expected T-step regret is bounded from above as

$$\begin{aligned}\mathbb{E}[R(T)] &= \Delta m + \Delta \sum_{t=2m+1}^{T} \mathbb{E}[\mathbb{1}\{j_t = 2\}] \\ &\leq \Delta m + \Delta \sum_{t=2m+1}^{T} \mathbb{E}[\mathbb{1}\{j_t = 2,\ \mathcal{E}\}] + 4T^2 \exp[-2\varepsilon^2 m] \,,\end{aligned}$$

where the last inequality follows from the definition of event $\mathcal{E}$ and observation that the maximum T-step regret is T. Let

$$m = \left\lceil \frac{16}{\widetilde{\Delta}^2} \log T \right\rceil \,, \quad \varepsilon = \frac{\widetilde{\Delta}}{4} \,,$$

where $\widetilde{\Delta}$ is a tunable parameter that determines the number of exploration steps per arm. From the definition of m and $\widetilde{\Delta}$, and the fact that $\mathbb{E}[\mathbb{1}\{j_t = 2,\ \mathcal{E}\}] = 0$ when $\widetilde{\Delta} \leq \Delta$, we have that

$$\mathbb{E}[R(T)] \leq \frac{16\Delta}{\widetilde{\Delta}^2} \log T + \widetilde{\Delta} T + \Delta + 4 \,.$$

Finally, note that $\Delta \leq 1$ and we choose $\widetilde{\Delta} = \left(\frac{16 \log T}{T}\right)^{\frac{1}{3}}$ that optimizes the upper bound.

C.3 Weighted bootstrapping and equivalence to TS

In this section, we prove that for the common reward distributions, WB becomes equivalent to TS for specific choices of the weight distribution and the transformation function.

C.3.1 Using multiplicative exponential weights

In this subsection, we consider multiplicative exponential weights, implying that $w_i \sim Exp(1)$ and $\mathcal{T}(y_i, w_i) = y_i \cdot w_i$. We show that in this setting WB is mathematically equivalent to TS for Bernoulli and more generally categorical rewards.

Proof for Proposition 1

Proof. Recall that the bootstrap sample is given as:

$$\widetilde{\theta} = \frac{\sum_{i=1}^{n}[w_i \cdot y_i] + \sum_{i=1}^{\alpha_0}[w_i]}{\sum_{i=1}^{n+\alpha_0+\beta_0} w_i}$$

To characterize the distribution of $\widetilde{\theta}$, let us define P_0 and P_1 as the sum of weights for the positive and negative examples respectively. Formally,

$$P_0 = \sum_{i=1}^{n} [w_i \cdot \mathcal{I}\{y_i = 0\}] + \sum_{i=1}^{\alpha_0}[w_i]$$
$$P_1 = \sum_{i=1}^{n} [w_i \cdot \mathcal{I}\{y_i = 1\}] + \sum_{i=1}^{\beta_0}[w_i]$$

The sample $\widetilde{\theta}$ can then be rewritten as:

$$\widetilde{\theta} = \frac{P_1}{P_0 + P_1}$$

Observe that P_0 (and P_1) is the sum of $\alpha + \alpha_0$ (and $\beta + \beta_0$ respectively) exponentially distributed random variables. Hence, $P_0 \sim Gamma(\alpha + \alpha_0, 1)$ and $P_1 \sim Gamma(\beta + \beta_0, 1)$. This implies that $\widetilde{\theta} \sim Beta(\alpha + \alpha_0, \beta + \beta_0)$.

When using the $Beta(\alpha_0, \beta_0)$ prior for TS, the corresponding posterior distribution on observing α positive examples and β negative examples is $Beta(\alpha + \alpha_0, \beta + \beta_0)$. Hence computing $\widetilde{\theta}$ according to WB is the equivalent to sampling from the Beta posterior. Hence, WB with multiplicative exponential weights is mathematically equivalent to TS. □

Categorical reward distribution

Proposition 3. *Let the rewards* $y_i \sim Multinomial(\theta_1^*, \theta_2^*, \dots \theta_C^*)$ *where* C *is the number of categories and* θ_i^* *is the probability of an example belonging to category* i*. In this case, weighted bootstrapping with* $w_i \sim Exp(1)$ *and the transformation* $y_i \to y_i \cdot w_i$ *results in* $\widetilde{\theta} \sim Dirichlet(n_1 + \widetilde{n}_1, n_2 + \widetilde{n}_2, \dots n_c + \widetilde{n}_c)$ *where* n_i *is the number of observations and* $\widetilde{n}_i$ *is the pseudo-count for category* i*. In this case, WB is equivalent to Thompson sampling under the* $Dirichlet(\widetilde{n}_1, \widetilde{n}_2, \dots \widetilde{n}_C)$ *prior.*

Proof. Like in the Bernoulli case, for all $c \in C$, define P_c as follows:

$$P_c = \sum_{i=1}^{n_c} [w_i \cdot \mathcal{I}\{y_i = c\}] + \sum_{i=1}^{\widetilde{n}_c} [w_i]$$

The bootstrap sample $\widetilde{\theta}$ consists of C dimensions i.e. $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2 \dots \widetilde{\theta}_C)$ such that:

$$\widetilde{\theta}_c = \frac{P_c}{\sum_{i=1}^{C} P_c}$$

Note that $\sum_{c=1}^{C} \widetilde{\theta}_c = 1$. Observe that P_c is the sum of $n_c + \widetilde{n}_c$ exponentially distributed random variables. Hence, $P_c \sim Gamma(n_c + \widetilde{n}_c, 1)$. This implies that $\widetilde{\theta} \sim Dirichlet(n_1 + \widetilde{n}_1, n_2 + \widetilde{n}_2 \dots n_k + \widetilde{n}_k)$.

When using the $Dirichlet(\widetilde{n}_1, \widetilde{n}_2, \dots \widetilde{n}_C)$ prior for TS, the corresponding posterior distribution is $Dirichlet(n_1 + \widetilde{n}_1, n_2 + \widetilde{n}_2 \dots n_k + \widetilde{n}_k)$. Hence computing $\widetilde{\theta}$ according to WB is the equivalent to sampling from the Dirichlet posterior. Hence, WB with multiplicative exponential weights is mathematically equivalent to TS. □

C.3.2 Using additive normal weights

In this subsection, we consider additive normal weights, implying that $w_i \sim N(0, 1)$ and $\mathcal{T}(y_i, w_i) = y_i + w_i$. We show that in this setting WB is mathematically equivalent to TS for Gaussian rewards.

Normal

Proposition 4. *Let the rewards $y_i \sim Normal(\langle \mathbf{x}_i, \theta^* \rangle, 1)$ where $\mathbf{x}_i$ is the corresponding feature vector for point i. If X is the $n \times d$ matrix of feature vectors and $\mathbf{y}$ is the vector of labels for the n observations, then weighted bootstrapping with $w_i \sim N(0,1)$ and using the transformation $y_i \to y_i + w_i$ results in $\widetilde{\theta} \sim N(\widehat{\theta}, \Sigma)$ where $\Sigma^{-1} = X^T X$ and $\widehat{\theta} = \Sigma \left[X^T \mathbf{y} \right]$. In this case, WB is equivalent to Thompson sampling under the uninformative prior $\theta \sim N(0, \infty)$.*

Proof. The probability of observing point i when the mean is θ and assuming unit variance,

$$\mathcal{P}(y_i | \mathbf{x}_i, \theta) = N(\langle \mathbf{x}_i, \theta \rangle, 1)$$

The log-likelihood for observing the data is equal to:

$$\mathcal{L}(\theta) = \frac{-1}{2} \sum_{i=1}^{n} (y_i - \langle x_i, \theta \rangle)^2$$

The MLE has the following closed form solution:

$$\widehat{\theta} = \left(X^T X \right)^{-1} X^T \mathbf{y}$$

The bootstrapped log-likelihood is given as:

$$\widetilde{\mathcal{L}}(\theta) = \frac{-1}{2} \sum_{i=1}^{n} (y_i + w_i - \langle x_i, \theta \rangle)^2$$

If $\mathbf{w} = [w_1, w_2 \ldots w_n]$ is the vector of weights, then the bootstrap sample can be computed as:

$$\widetilde{\theta} = \left(X^T X \right)^{-1} X^T \left[\mathbf{y} + \mathbf{w} \right]$$

The bootstrap estimator $\widetilde{\theta}$ has a Gaussian distribution since it is a linear combination of Gaussian random variables ($\mathbf{y}$ and $\mathbf{w}$). We now calculate the first and second moments for

$\widetilde{\theta}$ wrt to the random variables $\mathbf{w}$.

$$\begin{aligned}
\mathbb{E}[\widetilde{\theta}] &= \mathbb{E}_{\mathbf{w}}\left[\left(X^T X\right)^{-1} X^T \left[\mathbf{y} + \mathbf{w}\right]\right] \\
&= \left(X^T X\right)^{-1} X^T \mathbf{y} + \mathbb{E}\left[\left(X^T X\right)^{-1} X^T \mathbf{w}\right] \\
&= \widehat{\theta} + \left(X^T X\right)^{-1} X^T \mathbb{E}[\mathbf{w}] \\
\implies \mathbb{E}_{\mathbf{w}}[\widetilde{\theta}] &= \widehat{\theta}
\end{aligned}$$

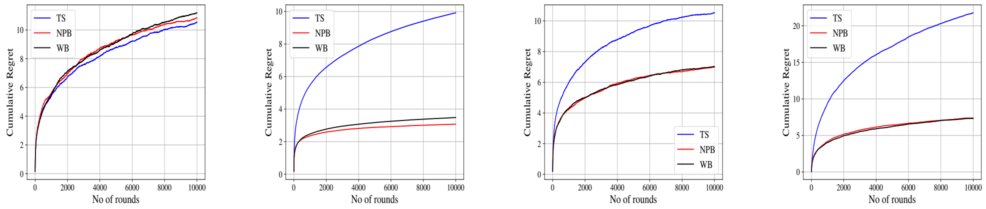
$$\begin{aligned}
\mathbb{E}_{\mathbf{w}}\left[(\widetilde{\theta} - \widehat{\theta})(\widetilde{\theta} - \widehat{\theta})^T\right] &= \mathbb{E}_{\mathbf{w}}\left[\left[(X^T X)^{-1} X^T \mathbf{w}\right]\left[(X^T X)^{-1} X^T \mathbf{w}\right]^T\right] \\
&= \mathbb{E}\left[\left[(X^T X)^{-1} X^T \mathbf{w}\mathbf{w}^T X (X^T X)^{-T}\right]\right] \\
&= \mathbb{E}\left[(X^T X)^{-1} X^T \mathbf{w}\mathbf{w}^T X (X^T X)^{-T}\right] \\
&= (X^T X)^{-1} X^T E\left[\mathbf{w}\mathbf{w}^T\right] X (X^T X)^{-T} \\
&= (X^T X)^{-1} X^T X (X^T X)^{-T} \qquad \text{(Since } \mathbb{E}[\mathbf{w}\mathbf{w}^T] = I_d\text{)} \\
&= (X^T X)^{-1} (X^T X) (X^T X)^{-1} \\
\implies \mathbb{E}_{\mathbf{w}}\left[(\widetilde{\theta} - \widehat{\theta})(\widetilde{\theta} - \widehat{\theta})^T\right] &= (X^T X)^{-1} = \Sigma
\end{aligned}$$

Thus $\widetilde{\theta} \sim N(\widehat{\theta}, \Sigma)$. When using the uninformative prior $N(0, \infty I_d)$ prior for TS, the posterior distribution on observing $\mathcal{D}$ is equal to $N(\widehat{\theta}, \Sigma)$. Hence computing $\widetilde{\theta}$ according to WB is the equivalent to sampling from the the Gaussian posterior. Hence, WB with additive normal weights is mathematically equivalent to TS. □

C.4 Additional Experimental Results

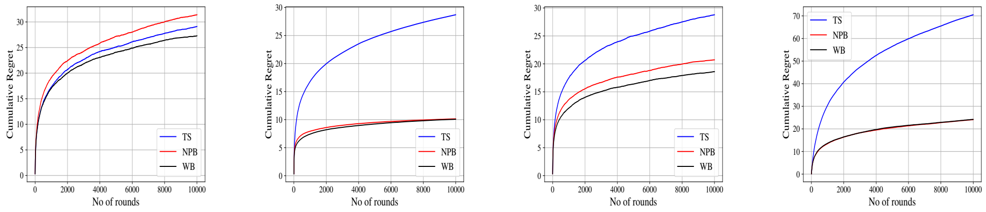
C.4.1 Bandit setting

C.4.2 Contextual bandit setting - Comparison to the method proposed in (McNellis et al., 2017)



(a) Bernoulli **(b)** Truncated-Normal **(c)** Beta **(d)** Triangular

Figure C.1: Cumulative Regret for TS, NPB and WB in a bandit setting $K = 2$ arms for (a) Bernoulli (b) Truncated-Normal in $[0, 1]$ (c) Beta (d) Triangular in $[0, 1]$ reward distributions

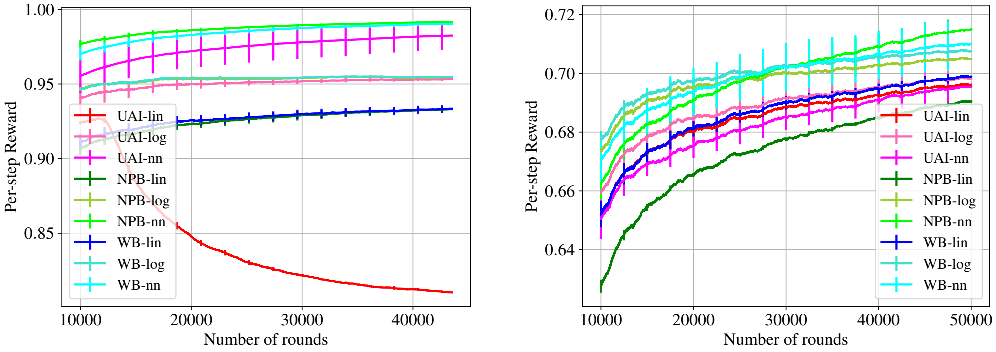


(a) Bernoulli **(b)** Truncated-Normal **(c)** Beta **(d)** Triangular

Figure C.2: Cumulative Regret for TS, NPB and WB in a bandit setting $K = 5$ arms for (a) Bernoulli (b) Truncated-Normal in $[0, 1]$ (c) Beta (d) Triangular in $[0, 1]$ reward distributions

Dataset	UAI-log	UAI-nn	NPB-log	NPB-nn	WB-log	WB-nn
Statlog	0.90	0.69	0.035	0.093	0.032	0.10
CovType	1.14	0.74	0.062	0.14	0.061	0.14

Table C.1: Runtime in seconds/round for non-linear variants of UAI, NPB and WB.



(a) Statlog

(b) CovType

Figure C.3: Comparison of the method proposed in (McNellis et al., 2017) (denoted as UAI in the plots), NPB and WB. The proposed bootstrapping methods tend to perform better than or equal to the method UAI. For UAI, we use an ensemble size of 5, 10 Gaussian feature-vectors as pseudo-examples and use the same stochastic optimization procedures as NPB and WB. In each round, we independently add the feature-vector and reward to a model in the ensemble independently with probability 0.5.