Sequential Learning with Similarities

Tomáš Kocák

École Doctorale Sciences pour l'Ingénieur
Inria Lille - Nord Europe
Université Lille 1

Thèse de Doctorat

présentée pour obtenir le grade de
DOCTEUR EN SCIENCES DE L'UNIVERSITÉ LILLE 1

Spécialité : **Informatique**

présentée par
Tomáš KOCÁK

Apprentissage séquentiel avec similitudes

---

sous la direction de M. Michal **VALKO**
et la co-direction de M. Rémi **MUNOS**

Rapporteurs: M. Claudio **GENTILE** University of Insubria
M. András **GYÖRGY** Imperial College London

Soutenue publiquement le **28 novembre 2016** devant le jury composé de :

M. Olivier	**CAPPÉ**	CNRS, Télécom ParisTech	Examinateur
M. Claudio	**GENTILE**	University of Insubria	Rapporteur
M. András	**GYÖRGY**	Imperial College London	Rapporteur
M. Rémi	**MUNOS**	Inria & Google DeepMind	Co-Directeur
M. Michal	**VALKO**	Inria Lille - Nord Europe	Directeur

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Michal Valko for the continuous support of my Ph. D study and related research, for his motivation, immense knowledge, great research insights and especially for his patience. His guidance helped me significantly in research and while writing all the papers and this thesis.

I would like to thank also my co-advisor Rémi Munos for his expert advice whenever needed and for his interesting and helpful discussions.

Besides my advisors, I would like to thank the rest of my thesis committee: Olivier Cappé, Claudio Gentile, and András György for accepting my invitation to the committee and especially my thanks go to Claudio Gentile and András György for accepting to review my thesis.

I also thank all the collaborators I worked with during my Ph. D study and all members SequeL team which contributed to the great experiance I had at Inria. Especially I thank Gergely Neu for countless hours discussing about our research.

Last but not the least, I would like to thank my family and friends for supporting me in difficult time and encouraging me to do my best.

Abstract

This thesis studies several extensions of multi-armed bandit problem, where a learner sequentially selects an action and obtains the reward of the action. Traditionally, the only information the learner acquires is about the obtained reward while information about other actions is hidden from the learner. This limited feedback can be restrictive in some applications like recommender systems, internet advertising, packet routing, etc. Usually, these problems come with structure, similarities between users or actions, additional observations, or any additional assumptions. Therefore, it is natural to incorporate these assumptions to the algorithms to improve their performance. This thesis focuses on multi-armed bandit problem with some underlying structure usually represented by a graph with actions as vertices. First, we study a problem where the graph captures similarities between actions; connected actions tend to grand similar rewards. Second, we study a problem where the learner observes rewards of all the neighbors of the selected action. We study these problems under several additional assumptions on rewards (stochastic, adversarial), side observations (adversarial, stochastic, noisy), actions (one node at the time, several nodes forming a combinatorial structure in the graph). The main contribution of this thesis is to design algorithms for previously mentioned problems together with theoretical and empirical guarantees. We also introduce several novel quantities, to capture the difficulty of some problems, like effective dimension and effective independence number.

Keywords: Sequential learning, bandit games, machine learning, decision making

Résumé

Dans cette thèse nous étudions différentes généralisations du problème dit « du bandit manchot ». Le problème du bandit manchot est un problème de décision séquentiel au cours duquel un agent sélectionne successivement des actions et obtient une récompense pour chacune d'elles. On fait généralement l'hypothèse que seule la récompense associée à l'action choisie est observée par l'agent, ce dernier ne reçoit aucune information sur les actions non choisies. Cette hypothèse s'avère parfois très restrictive pour certaines applications telles que les systèmes de recommandations, la publicité en ligne, le routage de paquets, etc. Ces types de problèmes sont en effet souvent très structurés : les utilisateurs et/ou les actions disponibles peuvent par exemple présenter certaines similitudes, ou l'agent peut parfois recevoir davantage d'information de l'environnement, etc. Il paraît dès lors assez naturel de tenir compte de la connaissance de la structure du problème pour améliorer les performances des algorithmes d'apprentissage usuels. Dans cette thèse, nous nous focalisons sur les problèmes de bandits présentant une structure pouvant être modélisée par un graphe dont les nœuds représentent les actions. Dans un premier temps, nous étudierons le cas où les arêtes du graphe modélisent les similitudes entre actions : deux actions connectées auront tendance à fournir des récompenses similaires. Dans un second temps, nous analyserons le cas où l'agent observe les récompenses de toutes les actions adjacentes à l'action choisie dans le graphe. Pour les deux cas précédents, nous dissocierons plusieurs sous-cas : récompenses stochastiques ou adversariales, informations additionnelles stochastiques adversariales ou bruitée, une ou plusieurs actions sélectionnées simultanément. Notre contribution principale a été d'élaborer de nouveaux algorithmes permettant de traiter efficacement les problèmes évoqués précédemment, et de démontrer théoriquement et empiriquement le bon fonctionnement de ces algorithmes. Nos travaux nous ont également amenés à introduire de nouvelles grandeurs, telles que la dimension effective et le nombre d'indépendance effectif, afin de caractériser la difficulté des différents problèmes.

Mots-clés: apprentissage séquentiel, jeux de bandits, apprentissage automatique, prise de décision (statistique)

List of author's related publications

T. Kocák, G. Neu, and M. Valko. Online learning with Erdős-Rényi side-observation graphs. In *Conference on Uncertainty in Artificial Intelligence*, 2016b

T. Kocák, G. Neu, and M. Valko. Online learning with noisy side observations. In *International Conference on Artificial Intelligence and Statistics*, 2016a

T. Kocák, G. Neu, M. Valko, and R. Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Neural Information Processing Systems*, 2014a

T. Kocák, M. Valko, R. Munos, and S. Agrawal. Spectral Thompson sampling. In *AAAI Conference on Artificial Intelligence*, 2014b

M. Valko, R. Munos, B. Kveton, and T. Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, 2014

T. Kocák, M. Valko, R. Munos, B. Kveton, and S. Agrawal. Spectral bandits for smooth graph functions with applications in recommender systems. In *AAAI Workshop on Sequential Decision-Making with Big Data*, 2014c

Contents

Chapter 1

Introduction

In this chapter, we introduce a sequential game called *multi-armed bandit* problem which plays the central role in this thesis. We first introduce the problem in both stochastic and adversarial environment and show several real world problems where multi-armed bandit problem can be applied. Later, we show some limitations of the approach of bandits in some real world scenarios and introduce several practical extensions of the problem studied in the past. The last part of this chapter introduces new extensions to the multi-armed bandit problem which try to tackle some limitations of traditional approaches. We focus mainly on the problems with some additional, and usually richer, structure while we aim for the solutions capturing the nature of the problems, and bringing theoretical and empirical improvements over already existing approaches.

1 Basic multi-armed bandits

The multi-armed bandit problem was originally inspired by clinical trials [Thompson, 1933]. In this problem, the doctor is facing a task to sequentially prescribe drugs to patients as they arrive. The aim of the doctor it to cure as many patients as possible. However, the problem received his name after another application of this framework. The terminology of multi-armed bandit problem is closely related to slot machines which are sometimes called one-armed bandits since the lever of a machine is also called arm. In this problem, the player faces several slot machines (several arms) and the player can choose one of the slot machines (one of the arms) to play and possibly receives a random reward. The goal of the player is to sequentially choose actions in order to earn as much as possible.

However, the random nature of the problem brings one important question. Suppose the learner already explored some of the actions and he finds one arm which tends to be the best so far. However, this might be caused by the randomness of the rewards.

Therefore, the learn faces a dilemma whether to play "the best" action or play some other action in order to gain more information and consequently make "better", more informed, decision. This problem is usually called exploration-exploitation dilemma and the main problem is to find a good balance between exploration and exploitation.

The multi-armed bandit problem can be applied in a wide variety of situations and has been studied in detail in the past. Now we show historical motivation for the problem as well as several applications of the problem in present days.

1.1 Motivation for multi-armed bandits

Clinical trials. The multi-armed bandit problem was initially motivated by clinical trials [Thompson, 1933] where the patients infected with a disease are treated by a set of drugs (one at the time). Effects of the drugs on infected patients are unknown at the beginning and the goal of the experiment is to find the best drug for the disease while curing as many people in the process as possible. An action or an arm, in this case, is a drug and a reward is whether we treated the patient successfully or not.

Packet routing. Consider a network represented by a set of vertices connected by edges and we want to send packets from the source to the destination in a given network. Every edge in a network is associated with an unknown delay depending on a traffic. In every trial a packet is sent along a chosen route from the source to the destination and the total delay of the packet is observed. The goal in this problem is to minimize the total delay of sending the packets. This problem was tackled by many papers including [Takimoto and Warmuth, 2003, McMahan and Blum, 2004, Awerbuch and Kleinberg, 2004, György et al., 2007] and several extensions of bandits were proposed to capture the problem more precisely. We will discuss these extensions later in this section.

Recommender systems. For many people recommender systems [Jannach et al., 2010] are integral parts of their lives. Watching movies, listening to the music, looking for a dinner recipe we, finding a good book or restaurant. All of these situations can be formalized as a bandit problem where a recommender system suggests an item to a user and receives a feedback (whether the user likes the item or not). Using this interaction, the system can learn user preferences in order to improve recommendations in the future.

Internet advertising. Consider a simple problem where an advertiser can show

you one ad from the set of possible ads [Pandey et al., 2007, Schwartz, 2013, Babaioff et al., 2014]. Every time you see an ad you can decide whether you want to click on it or not and the goal of the advertiser is to show you ads in order to maximize the number of clicks.

Sometimes the reward of an action can be simple and not changing too much over time (treatment for a disease) and sometimes rewards can change dramatically over time (the user suddenly started to like some movie genre). Therefore, the multi-armed bandit problem is studied under several feedback models. The most common are bandits with stochastic rewards where the rewards for an action come from a fixed distribution, and adversarial rewards where the rewards are chosen by an adversary without any statistical assumptions. We introduce these feedback models together with the setting and the goal of the learner in the following two sections.

1.2 Stochastic bandits

This problem was originally formulated by Robbins [1952]. The learner faces a set of N actions $\mathcal{A} = [N] \stackrel{\text{def}}{=} \{1, \ldots, N\}$. Each action i is associated with an unknown probability distribution ν_i on $[0, 1]$ with mean μ_i. At each time step $t \in [T]$, where $T \in \mathbb{N}$, the learner selects one action $a_t \in \mathcal{A}$ and receives a reward $r_t \sim \nu_{a_t}$ associated with the arm a_t. The goal of the learner is to maximize the expected reward he accumulates during the game; the sum of all the expected rewards $\sum_{t=1}^{T} \mathbb{E}[r_t] = \sum_{t=1}^{T} \mu_{a_t}$. Knowing the reward distributions the learner could play always the action $a_* \stackrel{\text{def}}{=} \arg\max_{i \in [N]} \mu_i$ with the highest expected reward. In order to analyze the performance of the learner, we compare his performance to the (optimal) strategy playing the best action in every time step. This performance measure is usually called *cumulative (pseudo) regret* denoted by R_T and defined as

$$R_T = T \max_{i \in [N]} \mu_i - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right],$$

where the expectation is taken with respect to the randomness of the adversary as well as with respect to the (possibly randomized) choices of the learner. Note that we measure the performance of the user in terms of cumulative regret instead of cumulative reward even though maximizing both of them leads to the same goal. Figure 1.1 summarizes the stochastic multi-armed bandit game.

Input:
  Known set of actions $[N]$
  Possibly known time horizon T
  Unknown probability distributions $\nu_1, \ldots, \nu_N$ such that $\mathbb{E}[\nu_i] = \mu_i, \ \forall i \in [N]$
for $t = 1$ **to** T **do**
  The learner chooses an action $a_t \in [N]$
  The learner receives a reward $r_t \sim \nu_{a_t}$
end for
Goal of the learner: Minimize cumulative regret $R_t = T \max_{i \in [N]} \mu_i - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right]$

Figure 1.1: Stochastic multi-armed bandit game

1.3 Adversarial bandits

Similarly to the stochastic case, the learner faces a set $\mathcal{A} = [N]$ of N actions and the game is played for $T \in \mathbb{N}$ rounds. However, the rewards associated with arms are not stochastic anymore. In each time step, an adversary privately assigns rewards $r_{t,i}$ to all the actions and the learner selects one action a_t to play. Then the learner receives a reward r_{t,a_t} corresponding to the action.

Similarly to the stochastic setting, the goal of the learner is to maximize its cumulative reward and thus, minimizing cumulative regret. Therefore, the performance of the learner is measured in terms of cumulative regret defined as

$$R_T = \max_{i \in [N]} \sum_{t=1}^{T} r_{t,i} - \sum_{t=1}^{T} r_{t,a_t},$$

which can be bounded either with high probability or in expectation. In the second case, cumulative pseudo-regret takes the form

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} \left(r_{t,i} - r_{t,a_t}\right)\right],$$

where the expectation is taken with respect to the randomness of the learner.

Input:
  Known set of actions $[N]$
  Possibly known time horizon T
  An adversary privately chooses rewards $r_{t,i}$ for all $i \in [N]$ and $t \in [T]$
for $t = 1$ **to** T **do**
  The learner chooses an action $a_t \in [N]$
  The learner receives a reward r_{t,a_t} corresponding to the action a_t
end for
Goal of the learner: Minimize cumulative regret R_T

Figure 1.2: Adversarial multi-armed bandit game

2 Extensions of multi-armed bandits

The formalism of multi-armed bandits can be easily used in the problems we mentioned before and in many other problems. However, actions in the multi-armed bandit problem are assumed to be independent and thus, provide no information about each other. On the other hand, real-world problems often come with some structure. Using this structure, one might be able to design an algorithm which can learn faster. For example, in packet routing, we usually observe delays on individual segments of the route. Moreover, the learner has also some information about the paths which share some sub-path with our chosen path. In recommender systems, the users usually like similar items similarly and in the internet advertising, users might be interested in a specific type of products like electronics, clothes, etc. Therefore, several extensions of multi-armed bandit problem have been studied in the past. We present several extensions in the following sections while focusing mainly on extensions related to the results presented in the thesis.

2.1 Full-information problem

Some problems come with much richer feedback than bandits. A good example is trading on a stock market where all stock prices are fully observable after each trading period. This can be formalized as a full-information problem (sometimes also called a problem of prediction with expert advice) [Vovk, 1990, Littlestone and Warmuth, 1994, Freund and Schapire, 1997, Cesa-Bianchi et al., 1997]. It is a sequential decision-making problem where, similarly to bandits, the learner picks an action and obtain the reward of the selected action. However, the main difference is that the

learner observes the losses associated with all potential decision, regardless of his choice. Even though the full-information problem has been studied independently of multi-armed bandit problem, the problems share many similarities. The standard algorithm for the problem is called Hedge with the optimal theoretical bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ where $\widetilde{\mathcal{O}}$ is a variation of $\mathcal{O}$ notation ignoring log factors. Using all additional information removes $\sqrt{N}$ factor from the optimal regret bound in the bandit case which is of $\widetilde{\mathcal{O}}(\sqrt{NT})$.

2.2 Linear and contextual multi-armed bandits

In linear bandits [Auer, 2002, Li et al., 2010, Agrawal and Goyal, 2013], every arm is associated with a D-dimensional vector (or a point in $\mathbb{R}^D$) and the reward function is an unknown linear function in $\mathbb{R}^D$. The problem can be also seen as learning an unknown D-dimensional vector $\boldsymbol{\alpha}$ such that the reward corresponding to an action is $\mathbf{x}^\intercal\boldsymbol{\alpha}$, where $\mathbf{x}$ is a vector corresponding to the action.

Contextual bandits bring very similar assumption on the rewards. Every action is associated with a possibly changing vector and the reward corresponding to an action can be obtained applying a function (unknown to the learner) on the vector. Usually the function is linear but not necessarily.

For these settings, Auer [2002] proposed SupLinRel algorithm and showed that it obtains $\widetilde{\mathcal{O}}(\sqrt{DT})$ regret, which matches the lower bound by Dani et al. [2008]. However, the first practical and empirically successful algorithm was LinUCB [Li et al., 2010]. Later, Chu et al. [2011] analyzed SupLinUCB, which is a LinUCB equivalent of SupLinRel. They showed that SupLinUCB also obtains $\widetilde{\mathcal{O}}(\sqrt{DT})$ regret. Abbasi-Yadkori et al. [2011] proposed OFUL for linear bandits which obtains $\widetilde{\mathcal{O}}(D\sqrt{T})$ regret. Using their analysis, it is possible to show that LinUCB obtains $\widetilde{\mathcal{O}}(D\sqrt{T})$ regret as well (Remark 6). Whether LinUCB matches the $\Omega(\sqrt{DT})$ lower bound for this setting is still an open problem.

Apart from the above approaches, an older approach to the problem is Thompson Sampling [Thompson, 1933]. Even though the algorithms based on Thompson Sampling are empirically very successful [Chapelle and Li, 2011], it took a long time to provide strong theoretical guarantees. Thompson Sampling for linear bandits was analyzed only recently, Agrawal and Goyal [2013] bring a new martingale technique which enabled them to show $\widetilde{\mathcal{O}}(D\sqrt{T})$ regret bound of LinearTS. Abernethy et al. [2008] and Bubeck et al. [2012] studied a more difficult *adversarial* setting of linear

bandits where the reward function is time-dependent. However, it is also an open problem if this approach has an upper bound on the regret that scales with $\sqrt{D}$, instead of D.

2.3 Combinatorial multi-armed bandits

A constraint on the number of arms played by the learner in the multi-armed bandit problem can present an issue in some applications, e.g. packet routing, where the action consists of picking several connections in the network forming a path. Combinatorial multi-armed bandits [Koolen et al., 2010, Cesa-Bianchi and Lugosi, 2012, Audibert et al., 2014] deal with this issue. It is a sequential problem where, in each time step t the environment assigns a loss value to each out of N components and the task of the learner is to choose one of the actions while trying to minimize the loss he incurs. Unlike in basic multi-armed bandit problem, where the actions consist of individual components, in combinatorial multi-armed bandit problem the actions can consist of several components. Usually, the action set $\mathcal{S}$ can be expressed as a subset of $\{0,1\}^N$ and playing an action $\boldsymbol{v} \in \mathcal{S}$ results in incurring loss of components corresponding to 1's in $\boldsymbol{v}$

3 Bandits with additional information

Real-world problems are usually complex with a rich structure. Therefore, a simple multi-armed bandit problem is usually not sufficient to capture the nature of the problem. On the other hand, extensions to the multi-armed bandit problem proved to be capable of capturing many different structures of the problems. However, there are still problems not captured by these extensions or sometimes algorithms just fail to capture the nature of the problem. For example, using a basic bandit algorithm to build a movie recommendation system comes with a problem. The algorithm needs to recommend every movie at least once which goes against the nature of the problem. We address this, and several other issues in this thesis.

In the rest of this chapter, we provide a brief overview of the bandit extensions studied in this thesis. We are mainly focusing on the problems with structure, usually represented by an underlying graph with actions as nodes. First, we look at the situation where the rewards of connected actions are correlated and thus, observing a reward of an action may give us some approximation of other correlated rewards. In

this extension, we aim for the algorithms which perform well only after a small number of steps; addressing the previously mentioned problem of recommender systems. Second, we explore the problem where the learner observes some additional information on top the reward of the action he plays. This additional information is in the form of a graph where playing an action reveals the rewards (possibly perturbed by noise) of the neighbors.

3.1 Spectral bandits and smooth graph functions

The first problem we study is called spectral bandits. It is a new problem which is motivated by a range of practical applications involving graphs. One application is *targeted advertisement* in social networks. Here, the graph is a social network and our goal is to discover a part of the network that is interested in a given product. Interests of people in a social network tend to change smoothly [McPherson et al., 2001], because friends tend to have similar preferences. Therefore, we take advantage of this structure and formulate this problem as learning a smooth preference function on a graph.

Another motivation for this approach are *recommender systems* [Jannach et al., 2010]. In content-based recommendation [Chau et al., 2011], the user is recommended items that are similar to the items that the user rated highly in the past. The assumption is that users prefer similar items similarly. The similarity of the items can be measured for instance by a nearest neighbor graph [Billsus et al., 2000], where each item is a node and its neighbors are the most similar items.

Our goal is to design algorithms which can leverage the fact that the reward function can be smooth in many applications and provide strong theoretical guarantees and empirical performance. Especially, we are aiming for algorithms that can perform well only after few time steps.

A smooth graph function is a function on a graph that returns similar values on neighboring nodes. This concept arises frequently in manifold and semi-supervised learning [Zhu, 2008], and reflects the fact that the outcomes on the neighboring nodes tend to be similar. It is well-known [Belkin et al., 2006, 2004] that a smooth graph function can be expressed as a linear combination of the eigenvectors of the graph Laplacian with smallest eigenvalues. Therefore, the problem of learning such function can be cast as a regression problem on these eigenvectors. We bring this concept to bandits. In particular, in Chapter 2 we study a bandit problem where the arms are

the nodes of a graph and the expected payoff of pulling an arm is a smooth function on this graph. We call this problem *spectral bandits*.

In the spectral bandit setting, we consider the following. The graph is known in advance and its edges represent the similarity of the nodes. At time t, we choose a node and then observe its payoff. In targeted advertisement, this may correspond to showing an ad and then observing whether the person clicked on the ad. In content-based recommendation, this may correspond to recommending an item and then observing the assigned rating. Based on the payoff, we update our model of the world and then the game proceeds into time $t + 1$. In both applications described above, the learner (advertiser) has rarely the budget (time T) to try all the options even once. Furthermore, imagine that the learner is a movie recommender system and would ask the user to rate all the movies before it starts producing relevant recommendations. Such a recommender system would be of little value. Yet, many bandit algorithms start with pulling each arm once. This is something that we cannot afford here and therefore, contrary to standard bandits, we mostly consider the case $T \ll N$, where the number of nodes N can be huge. While we are mostly interested in the regime when $T < N$, our results are beneficial also for $T \geq N$. This regime is especially challenging since traditional multi-arm bandit algorithms need to try every arm at least once.

If the smooth graph function can be expressed as a linear combination of k eigenvectors of the graph Laplacian, and k is small and known, our learning problem can be solved using ordinary linear bandits [Auer, 2002, Li et al., 2010, Agrawal and Goyal, 2013]. In practice, k is problem specific and unknown. Moreover, the number of features k may approach the number of nodes N. Therefore, proper regularization is necessary, so that the regret of the learning algorithm does not scale with N. We are interested in the setting where the regret is independent of N and this makes the problem we study non-trivial.

Later in Chapter 2 we make several major contributions. First, we formalize a bandit problem, where the payoff of the arms is a smooth function on a graph. Second, we introduce an *effective dimension* d which characterizes the hardness of the problem. Later we propose three algorithms for solving this problem that achieve regret bounds scaling with $d\sqrt{T}$ or $\sqrt{dT}$. Note that the regret bounds scale with the effective dimension d instead of ambient dimension $D = N$ like in linear bandits. Therefore we can expect improvement over linear bandits whenever d is smaller than D. This is reflected in the last part of the Chapter 2 where we evaluated the algorithms on both synthetic and real-world content-based recommendation problems.

3.2 Bandits with side observations

As we mentioned earlier, bandit feedback (observing only the reward of a selected action) can be too restrictive in some applications. The simplest problem with richer feedback is the previously mentioned setting with full information. However, the problems might be more complicated than observing one or all rewards. Therefore, we consider sequential decision-making problems where the feedback interpolates between full-information feedback [Koolen et al., 2010] and bandit feedback [Auer et al., 2002a]. This enables us to study practical, usually more complex problems where traditional approaches may lead to suboptimal results. Recently, Mannor and Shamir [2011] proposed a partial feedback scheme that models situations that lie between the two extremes: in their model, the learner observes losses associated with some additional actions besides its own loss. More precisely, there is an underlying graph structure with arms as nodes and playing an action also reveals the losses of all the neighbors according to the graph. We call this scheme *bandits with side observations*.

Later in Chapter 3 we study several settings derived from the setting of Mannor and Shamir [2011]. For each of the setting, we present an algorithm together with theoretical guarantees. Chapter 3 is divided into following parts.

Adversarial bandits with adversarial side observations. This is a basic setting of Mannor and Shamir [2011] where an underlying graph is selected by an adversary in every round. For this setting, we introduce an implicit exploration technique and present Exp3-IX (which uses implicit exploration). The implicit exploration technique enables us to prove the optimal regret bound for the algorithm even without access to the graph before playing an action (Exp3-IX is the first algorithm for directed graphs which does not need access to the graph before an action is played).

Adversarial bandits with stochastic side observations. In this setting, we assume that an underlying graph is constructed as an Erdős-Rényi graph with some parameter r. This parameter is selected by an adversary and is unknown to the learner. For this setting, we present the Exp3-Res algorithm which uses geometric resampling in order to utilize a limited number of side observations. This approach enables us to prove an optimal regret bound for the algorithm even without observing the underlying graph structure, not even after an action is played.

Adversarial bandits with noisy side observations. In this setting, we assume that an underlying graph is weighted and selected by an adversary. The weights

represent the amount of information contained in the feedback. A weight close to one means that the side observation is close to the real reward while a weight close to zero means that the side observation is almost pure noise. For this setting, we introduce a new quantity called *effective independence number* and present two algorithms together with theoretical guarantees. The first algorithm is called Exp3-IXt and it needs to know the graph beforehand in order to achieve optimal regret rate. The second algorithm is called Exp3-WIX and uses a novel type of loss estimates in order to achieve optimal regret bound, even without knowing the graph before taking an action.

Combinatorial semi-bandits with adversarial side observations In this setting we assume that the action of the learner is more complex than simply playing one node of the graph. Instead, the action consists of several nodes (e.g. path from one node to another, circles in the graph, pairs of nodes etc.), usually some combinatorial structure but it can be any subset of nodes. These actions are problem dependent. Similarly to previous settings, an adversary constructs a graph and playing an action (set of nodes) reveal also rewards of nodes connected to the action (connected to at least one node in the selected action).

Chapter 2

Spectral bandits for smooth graph functions

Smooth functions on graphs have wide applications in the manifold and semi-supervised learning. In this chapter, we study a bandit problem where the payoffs of arms are smooth on a graph. This framework is suitable for solving online learning problems that involve graphs, such as content-based recommendation. In this problem, each item we can recommend is a node and its expected rating is similar to its neighbors. The goal is to recommend items that have high expected ratings. We aim for the algorithms where the cumulative regret with respect to the optimal policy would not scale poorly with the number of nodes. In particular, we introduce the notion of an *effective dimension*, which is small in real-world graphs, and propose three algorithms for solving our problem that scales linearly and sublinearly in this dimension. Our experiments on content recommendation problem show that a good estimator of user preferences for thousands of items can be learned from just tens of node evaluations.

Contents

1 Introduction

A smooth graph function is a function on a graph that returns similar values on neighboring nodes. This concept arises frequently in manifold and semi-supervised learning [Zhu, 2008], and reflects the fact that the outcomes on the neighboring nodes tend to be similar. It is well-known [Belkin et al., 2006, 2004] that a smooth graph function can be expressed as a linear combination of the eigenvectors of the graph Laplacian with smallest eigenvalues. Therefore, the problem of learning such function can be cast as a regression problem on these eigenvectors. This work brings this concept to bandits. In particular, we study a bandit problem where the arms are

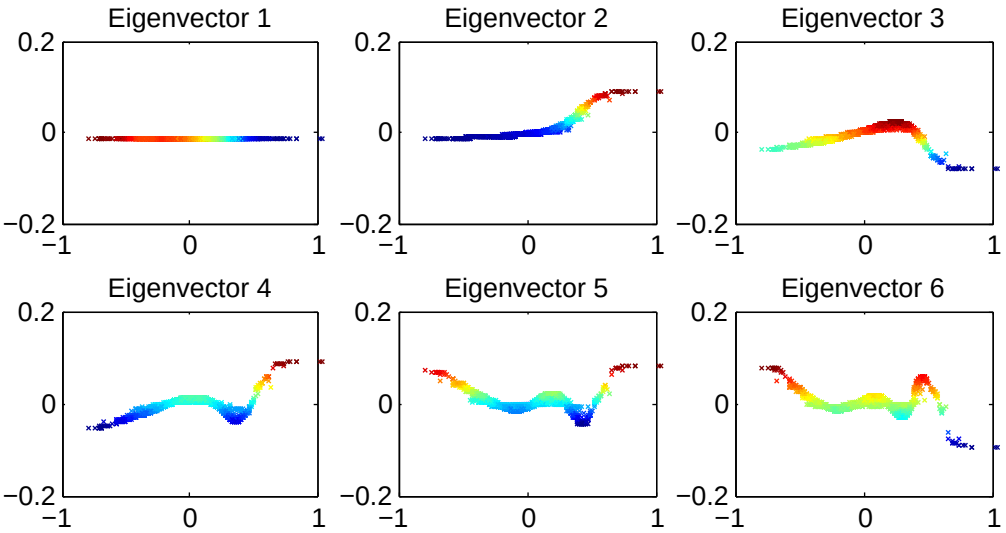


Figure 2.1: Eigenvectors from the Flixster data corresponding to the smallest few eigenvalues projected onto the first principal component. Colors indicate the values.

the nodes of a graph and the expected payoff of pulling an arm is a smooth function on this graph.

We are motivated by a range of practical problems that involve graphs. One application is *targeted advertisement* in social networks. Here, the graph is a social network and our goal is to discover a part of the network that is interested in a given product. Interests of people in a social network tend to change smoothly [McPherson et al., 2001] because friends tend to have similar preferences. Therefore, we take advantage of this structure and formulate this problem as learning a smooth preference function on a graph.

Another application of our work are *recommender systems* [Jannach et al., 2010]. In the content-based recommendation [Chau et al., 2011], the user is recommended items that are similar to the items that the user rated highly in the past. The assumption is that users prefer similar items similarly. The similarity of the items can be measured for instance by a nearest neighbor graph [Billsus et al., 2000], where each item is a node and its neighbors are the most similar items.

In this chapter, we consider the following learning setting. The graph is known in advance and its edges represent the similarity of the nodes. At time t, we choose a node and then observe its payoff. In targeted advertisement, this may correspond to showing an ad and then observing whether the person clicked on the ad. In content-based recommendation, this may correspond to recommending an item and

then observing the assigned rating. Based on the payoff, we update our model of the world and then the game proceeds into time $t+1$. Since the number of nodes N can be huge, we are mostly interested in the regime when $t < N$ even though our results are beneficial also for $t > N$. This regime is especially challenging since traditional multi-arm bandit algorithms need to explore every arm at least once.

If the smooth graph function can be expressed as a linear combination of k eigenvectors of the graph Laplacian, and k is small and known, our learning problem can be solved using ordinary linear bandits [Auer, 2002, Li et al., 2010, Agrawal and Goyal, 2013]. In practice, k is problem specific and unknown. Moreover, the number of features k may approach the number of nodes N. Therefore, proper regularization is necessary, so that the regret of the learning algorithm does not scale with N. We are interested in the setting where the regret is independent of N and therefore this problem is non-trivial.

2 Spectral bandit setting

In this section, we formally define the spectral bandit setting. Let $\mathcal{G}$ be the given graph with the set of nodes $\mathcal{V}$ and denote $|\mathcal{V}| = N$ the number of nodes. Let $\mathcal{W}$ be the $N \times N$ matrix of similarities w_{ij} (edge weights) and $\mathcal{D}$ is the $N \times N$ diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$ (node degrees). The graph Laplacian of $\mathcal{G}$ is defined as $\mathcal{L} = \mathcal{D} - \mathcal{W}$. Let $\{\lambda_k^{\mathcal{L}}, \mathbf{q}_k\}_{k=1}^N$ be the eigenvalues and eigenvectors of $\mathcal{L}$ ordered such that $0 = \lambda_1^{\mathcal{L}} \leq \lambda_2^{\mathcal{L}} \leq \cdots \leq \lambda_N^{\mathcal{L}}$. Equivalently, let $\mathcal{L} = \mathbf{Q}\mathbf{\Lambda}_{\mathcal{L}}\mathbf{Q}^\intercal$ be the eigendecomposition of $\mathcal{L}$, where $\mathbf{Q}$ is an $N \times N$ *orthogonal* matrix with eigenvectors in columns.

Eigenvectors of the graph Laplacian form a basis (principal axis theorem), therefore we can represent any reward function as a linear combination of the eigenvectors. For any set of weights $\boldsymbol{\alpha}$ let $f_{\boldsymbol{\alpha}} : \mathcal{V} \to \mathbb{R}$ be the reward function defined on nodes, linear in the basis of the eigenvectors of $\mathcal{L}$:

$$f_{\boldsymbol{\alpha}}(v) = \sum_{k=1}^{N} \alpha_k (\mathbf{q}_k)_v = \mathbf{x}_v^\intercal \boldsymbol{\alpha},$$

where $\mathbf{x}_v$ is the v-th row of $\mathbf{Q}$, i.e., $(\mathbf{x}_v)_i = (\mathbf{q}_i)_v$. If the weight coefficients of the true $\boldsymbol{\alpha}$ are such that the large coefficients correspond to the eigenvectors with the small eigenvalues and vice versa, then $f_{\boldsymbol{\alpha}}$ would be a smooth function on $\mathcal{G}$ [Belkin et al.,

2006]. For more details see Section 3.1. Figure 2.1 displays first few eigenvectors of the Laplacian constructed from the data we use in our experiments. In the extreme case, the true $\boldsymbol{\alpha}$ may be of the form $[\alpha_1, \alpha_2, \ldots, \alpha_k, 0, 0, 0]_N^\intercal$ for some $k \ll N$. Had we known k in such case, the known linear bandits algorithm would work with the performance scaling with k instead of $D = N$. Unfortunately, first, we do not know k and second, we do not want to assume such an extreme case (i.e., $\alpha_i = 0$ for $i > k$). Therefore, we opt for the more plausible assumption that the coefficients with the high indexes are small. Consequently, we deliver algorithms with the performance that scale with the smoothness with respect to the graph.

The learning setting is the following. In each time step $t \leq T$, the recommender chooses a node a_t and obtains a noisy reward such that:

$$r_t = \mathbf{x}_{a_t}^\intercal \boldsymbol{\alpha} + \varepsilon_t,$$

where the noise ε_t is assumed to be R-sub-Gaussian (i.e. $\mathbb{E}[\varepsilon_t] = 0$ and $\mathbb{E}[\exp(s\varepsilon_t)] \leq \exp(R^2 s^2/2)$, for all $s \in \mathbb{R}$) for any t. In our setting, we have $\mathbf{x}_v \in \mathbb{R}^D$ and $\|\mathbf{x}_v\|_2 \leq 1$ for all $\mathbf{x}_v$. The goal of the recommender system is to minimize the cumulative regret with respect to the strategy that always picks the best node w.r.t. $\boldsymbol{\alpha}$. Let a_t be the node picked (referred to as *pulling an arm*) by an algorithm at time t. The cumulative (pseudo) regret of the algorithm is defined as:

$$R_T = T \max_v f_{\boldsymbol{\alpha}}(v) - \sum_{t=1}^{T} f_{\boldsymbol{\alpha}}(a_t)$$

We call this bandit setting *spectral* since it is built on the spectral properties of a graph. Compared to the linear and multi-arm bandits, the number of arms K is equal to the number of nodes N and to the dimension of the basis D (eigenvectors are of dimension N). However, a regret that scales with N or D that can be obtained using those settings is not acceptable because the number of nodes can be large. While we are mostly interested in the setting with $K = N$, our algorithms and analyses can be applied for any finite K.

2.1 Related work

We are mostly interested in smooth graph functions in spectral bandit setting which can be expressed as a linear combination of eigenvectors of the graph Laplacian

(Chapter 2). Therefore, the most related settings to our work are that of the linear and contextual linear bandits (Section 2.2 in Chapter 1).

Kleinberg et al. [2008], Slivkins [2009], and Bubeck et al. [2011] use similarity information between the context of arms, assuming a Lipschitz or more general properties. While such settings are indeed more general, the regret bounds scale worse with the relevant dimensions. Srinivas et al. [2010] and Valko et al. [2013] also perform maximization over the smooth functions that are either sampled from a Gaussian process prior or have a small RKHS norm. Their setting is also more general than ours since it already generalizes linear bandits. However, their regret bound in the linear case scales with D. Moreover, the regret of these algorithms also depends on a quantity for which data-independent bounds exist only for some kernels, while our effective dimension is always computable given the graph.

Another bandit graph setting called the *gang of bandits* was studied by Cesa-Bianchi et al. [2013] where each node is a linear bandit with its own weight vector which is assumed to be smooth on the graph. Gentile et al. [2014] take a different approach to similarities in social networks by assuming that the actions are clustered into several unknown clusters and the actions within one cluster have the same expected reward. This approach can be applied also to the setting presented in our paper. The biggest advantage of the CLUB algorithm presented in Gentile et al. [2014] is that it constructs a graph iteratively, starting with the complete graph and removing edges which are not likely to be present in the underlying clustering. Therefore, the algorithm does not need to know the similarity graph unlike in our setting. However, theoretical improvement of CLUB compared to the basic bandit algorithm comes from the small number of clusters. Therefore, if the number of clusters is close to the number of actions the algorithm does not bring any improvement while the algorithms in our setting still can leverage the similarity structure. Li et al. [2015] later extended the approach to *double-clustering* where both the users and the items are assumed to appear in clusters (with the underlying clustering unknown to the learner) and Korda et al. [2016] considers a distributed extension. Yet another assumption of a special graph reward structure is exploited by unimodal bandits [Yu and Mannor, 2011, Combes and Proutière, 2014]. One of the settings considered by Yu and Mannor [2011] is a graph bandit setting where every path in the graph has unimodal rewards and therefore also imposes a specific kind of smoothness with respect to the graph topology. In networked bandits [Fang and Tao, 2014], the learner picks a node, but besides receiving the reward from that node, its reward is the sum of the rewards of the picked node and its neighborhood. The algorithm of Fang and Tao [2014], NetBandits, can also deal with changing topology, however, this has to be always

revealed to the learner before it makes its decision.

Spectral bandits with different objectives In the follow-up work on spectral bandits, there have been algorithms optimizing other objective functions than the cumulative regret. First, in some sensor networks, sensing a node (pulling and arm) has an associated cost [Narang et al., 2013]. In a particular, *cheap bandit* setting [Hanawal et al., 2015], it is cheaper to get an average of rewards of a set of nodes than a specific reward of a single one. More precisely, the learner has a fixed budget and pays the cost for the action which depends on the spectral properties of the graph while relying on the property that getting the average reward of many nodes is less costly than getting a reward of a single node. For this setting, Hanawal et al. [2015] proposed CheapUCB that reduces the cost of sampling by $1/4$ as compared to SpectralUCB, while maintaining $\tilde{\mathcal{O}}(d\sqrt{T})$ cumulative regret. Next, Gu and Han [2014] study the online classification setting on graphs with bandit feedback, very similar to spectral bandits; after predicting the class the oracle will return a single bit indicating whether the prediction is correct or not. The analysis of their algorithm delivers essentially the same bound on the regret, however, they need to know the number of relevant eigenvectors d. Moreover, Ma et al. [2015] consider several variants of Σ-*optimality* that favors specific exploration when selecting the nodes, for example, the learner is not allowed to play one arm multiple times.

3 Spectral bandits

In this section, we show how to leverage the smoothness of the rewards on a given graph. Thinking that the reward observed for an arm does not provide any information for other arms would not be correct because of the assumption that under another basis, the unknown parameter has a low norm. This provides an additional information across the arms through the estimation of the parameter $\boldsymbol{\alpha}$.

3.1 Smooth graph functions

There are several possible ways to define the *smoothness* of the function f with respect to the graph $\mathcal{G}$. We are using the one which is standard in spectral clustering

[Luxburg, 2007] and semi-supervised learning [Belkin et al., 2006], defined as:

$$S_G(f) = \frac{1}{2} \sum_{i,\, j \in [N]}^{N} w_{i,j} \left(f(i) - f(j)\right)^2 .$$

Therefore, whenever the function values of the nodes connected by an edge with large weight are close, the smoothness of the function with respect to the graph is small, and the function is smoother with respect to the graph. This definition has several useful properties. We are mainly interested in the following:

$$S_G(f) = \mathbf{f}^\intercal \mathcal{L} \mathbf{f} = \mathbf{f}^\intercal \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\intercal \mathbf{f} = \boldsymbol{\alpha}^\intercal \mathbf{\Lambda} \boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}^2 = \sum_{i=1}^{N} \lambda_i \alpha_i^2,$$

where $\mathbf{f} = (f(1), \ldots, f(N))^\intercal$ is the vector of the function values, $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\intercal$ is the eigendecomposition of the graph laplacian $\mathcal{L}$, and $\boldsymbol{\alpha} = \mathbf{Q}^\intercal \mathbf{f}$ is the representation of the vector $\mathbf{f}$ in the eigenbasis. The assumption on the smoothness of the reward function with respect to the underlying graph is reflected by the small value of $S_{\mathcal{G}}(f)$ and therefore, the components of $\boldsymbol{\alpha}$ corresponding to the large eigenvalues should be small as well.

As a result, we can think of our setting as a linear bandit problem with dimension N which is possibly larger than the time horizon T and the mean reward $f(k)$ for each arm k satisfies the property that under a change of coordinates, the vector $\mathbf{f}$ of mean rewards has small components, i.e., there exists a known orthogonal matrix $\mathbf{U}$ such that $\boldsymbol{\alpha} = \mathbf{U}\mathbf{f}$ has a low norm. As a consequence, we can estimate $\boldsymbol{\alpha}$ using penalization corresponding to the large eigenvalues and to recover $\mathbf{f}$.

Given a vector of weights $\boldsymbol{\alpha}$, we define its $\mathbf{\Lambda}$-norm as:

$$\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}} = \sqrt{\sum_{i=1}^{N} \lambda_i \alpha_i^2} = \sqrt{\boldsymbol{\alpha}^\intercal \mathbf{\Lambda} \boldsymbol{\alpha}}. \tag{2.1}$$

In fact, this norm is defined as the square root to the smoothness of the function and we utilize it later in our algorithms by regularization which enforces small $\mathbf{\Lambda}$-norm of $\boldsymbol{\alpha}$.

3.2 Effective dimension

In order to present and analyze our algorithms, we use a notion of *effective dimension* denoted by (lower case) d. This quantity was introduced in Valko et al. [2014] however, we use a slightly modified version of the effective dimension. This new definition of the effective dimension enables us to prove stronger theoretical guarantees for our algorithms. In the rest of the paper, we refer to the old definition of the effective dimension, introduced in Valko et al. [2014], as d_{old}. We keep using capital D to denote the ambient dimension (the number of features, the same as the number of actions in our setting). Intuitively, the effective dimension is a proxy for the number of relevant dimensions. We first provide a formal definition and then discuss its properties, including $d < d_{\text{old}} \ll D$.

In general, we assume to have an eigendecomposition $\mathcal{L} = \mathbf{Q}\mathbf{\Lambda}_{\mathcal{L}}\mathbf{Q}^{\intercal}$ of the graph Laplacian $\mathcal{L}$ and a diagonal matrix of regularized eigenvalues $\mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$ with the entries $0 < \lambda = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ for some $\lambda > 0$. Note that the smallest eigenvalue of the graph Laplacian is always zero. We use regularized eigenvalues in order to prevent dividing by zero. This enables us to define the effective dimension while still being able to control an error introduced by the regularization and therefore not spoiling the bounds for the spectral setting.

Definition 1. *Let the* ***effective dimension*** d *be defined as:*

$$d = \left\lceil \frac{\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil,$$

where the maximum is taken over all possible non-negative real numbers $\{t_1, \ldots, t_N\}$*, such that* $\sum_{i=1}^{N} t_i = T$ *and* K *is the number of zero eigenvalues (before regularization).*

Remark 1. *Note that if we first upper bound every* $1/\lambda_i$ *in the numerator by* $1/\lambda$ *then the maximum is acquired for* t_i *equal to* T/N*. Therefore, the right hand side of the definition can be bounded from above by* $D = N$*. This means that* d *is upper bounded by* D*. Later we show that in many practical situations,* d *is much smaller then* D*.*

For the comparison, we show also the previous definition of the effective dimension [Valko et al., 2014] and from now we will call it *old effective dimension* denoted by d_{old}.

Definition 2 (Old effective dimension [Valko et al., 2014]). *Let the* ***old effective dimension*** d_{old} *be the largest* $d_{old} \in [N]$ *such that:*

$$(d_{old} - 1)\lambda_{d_{old}} \leq \frac{T}{\log(1 + T/\lambda)}$$

Remark 2. *Note that from Lemma 5 and Lemma 6 by Valko et al. [2014], we see that the relation between the old and new definition of the effective dimension is:* $d \leq 2d_{old}$. *As we show later, the bounds using the effective dimension scale either with* d *or with* $2d_{old}$. *Moreover, we show that* d *is usually much smaller than* $2d_{old}$ *and therefore using the new definition of the effective dimension can bring an improvement to the bound.*

The effective dimension d is small when the coefficients λ_i grow rapidly above T. This is the case when the dimension of the space D is much larger than T, such as in graphs from social networks with a very large number of nodes N. In contrast, when the coefficients λ_i are all small (if the graph is sparse, all eigenvalues of Laplacian are small) then d may be of the order of T, which would make the regret bounds useless.

The actual form of Definition 1 comes from Lemma 11 and will become apparent in Section 5. The dependence of the effective dimension on T comes from the fact, that d is related to the number of "non-negligible" dimensions characterizing the space where the solution to the penalized least-squares may lie, since this solution is basically constrained to an ellipsoid defined by the inverse of the eigenvalues. This ellipsoid is wide in the directions corresponding to the small eigenvalues and narrow in the directions corresponding to the large ones. After playing an action, the confidence ellipsoid shrinks in the directions of the action. Therefore, exploring in a direction where the ellipsoid is wide can reduce the volume of the ellipsoid much more than exploring in a direction where the ellipsoid is narrow. In fact, for a small T, the axes of the ellipsoid corresponding to the large eigenvalues of $\mathcal{L}$ are negligible. Consequently, d is related to the metric dimension of this ellipsoid. Therefore, when T goes to infinity, all the directions matter, thus the solution can be anywhere in a (bounded) space of dimension N. On the contrary, for a smaller T, the ellipsoid possesses a smaller number of "non-negligible" dimensions.

3.2.1 The computation of the effective dimension

All of the algorithms that we propose need to know the value of the effective dimension in order to leverage the structure of the problem. Therefore, it is necessary to compute it beforehand. Usually, we proceed in two steps when computing the effective dimension:

1. Finding an N-tuple $(t_1, \ldots, t_N)$ which maximizes the expression from the definition of the effective dimension.

2. Plugging the N-tuple to the definition of the effective dimension.

Now we focus on the first step. The following lemma gives us an efficient way to determine the N-tuple

Lemma 1. *Let $\omega \in [N]$ be the largest integer such that*

$$\frac{\sum_{i=1}^{\omega} \lambda_i}{\omega} + \frac{T}{\omega} - \lambda_\omega > 0,$$

then $t_1, \ldots, t_N$ that maximize the expression in the definition of the effective dimension are in the following form:

$$\begin{aligned} t_i &= \frac{\sum_{i=1}^{\omega} \lambda_i}{\omega} + \frac{T}{\omega} - \lambda_i && \text{for } i = 1, \ldots, \omega, \\ t_i &= 0 && \text{for } i = \omega + 1, \ldots, N. \end{aligned}$$

Proof. First of all, we use the fact that logarithm is an increasing function and that the N-tuple which maximizes the expression is invariant to a multiplication of the expression by a constant:

$$\arg\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right) = \arg\max \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right) = \arg\max \prod_{i=1}^{N} (\lambda_i + t_i)$$

The last expression is easy to maximize since we know that for any $\Delta \geq \delta \geq 0$ and

for any real number a we have

$$\begin{aligned} 0 &\leq \Delta^2 - \delta^2 \\ a^2 - \Delta^2 &\leq a^2 - \delta^2 \\ (a-\Delta)(a+\Delta) &\leq (a-\delta)(a+\delta). \end{aligned}$$

Therefore, if we take any two terms $(\lambda_i + t_i)$ and $(\lambda_j + t_j)$ from the expression which we are maximizing, we can potentially increase their product simply by balancing them:

$$\begin{aligned} t_i^{\text{new}} &= \frac{\lambda_i + \lambda_j + t_i + t_j}{2} - \lambda_i \\ t_j^{\text{new}} &= \frac{\lambda_i + \lambda_j + t_i + t_j}{2} - \lambda_j. \end{aligned}$$

However, we still have to take into consideration that every t_i has to be positive. Therefore, if for example t_j^{new} is negative, we can simply set

$$\begin{aligned} t_i^{\text{new}} &= t_i + t_j \\ t_j^{\text{new}} &= 0. \end{aligned}$$

We can apply this argument to the expression we are trying to maximize to obtain the statement of the lemma. □

The second part is straightforward. To avoid computational difficulties of multiplying N numbers, we use properties of logarithm to get:

$$d = \left\lceil \frac{\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = \left\lceil \frac{\max \sum_{i=1}^{N} \log \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil.$$

Knowing an N-tuple which maximizes the expression, we can simply plug it in and obtain the value of the effective dimension.

3.2.2 The old vs. new definition of the effective dimension

As we mentioned in Remark 2, our new effective dimension is always upper bounded by $2d_{\text{old}}$. In this section, we show that the gap between d and $2d_{\text{old}}$ can be significant what we demonstrate on the graphs constructed for several real-world datasets, and also on several artificial graphs.

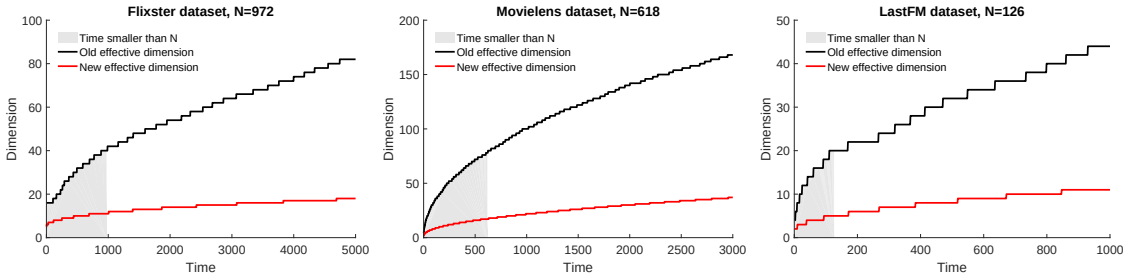


Figure 2.2: Difference between d and $2d_{\text{old}}$ for real world datasets. From left to right: Flixster dataset with $N = 972$, Movielens dataset with $N = 618$, and LastFM dataset with $N = 804$.

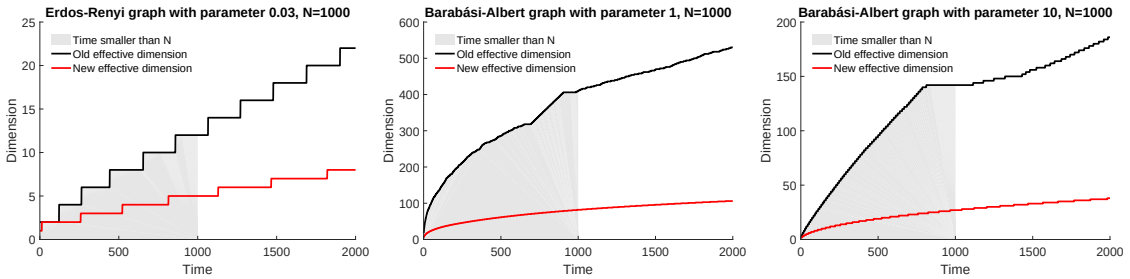


Figure 2.3: Difference between d and $2d_{\text{old}}$ for artificial graphs on $N = 1000$ vertices. From left to right: Erdős-Renyi graph with the probability 0.03 of an edge, Barabási-Albert graph with one edge per added vertex, Barabási-Albert graph with ten edges per added vertex.

Figures 2.2 and 2.3 show how d behaves compared to $2d_{\text{old}}$ on the generated and the real Flixster, Movielens, and LastFM network graphs.[1] We use some of them for the experiments in Section 6. The figures clearly demonstrate the gap between d and $2d_{\text{old}}$ while both of the quantities are much smaller then D. In fact, effective dimension d is much smaller than D even for $T > N$ (Figures 2.2 and 2.3). Therefore,

[1] We set $\mathbf{\Lambda}$ to $\mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$ with $\lambda = 0.1$, where $\mathbf{\Lambda}_{\mathcal{L}}$ is the graph Laplacian of the respective graph.

spectral bandits can be use even for $T > N$ while maintaining the advantage of better regret bound compared to the linear bandit algorithms.

3.3 Lower bound

In this section, we show a lower bound for the spectral setting. More precisely, for each possible value of effective dimension d and time horizon T, we show the existence of a "hard" problem with a lower bound of $\Omega(\sqrt{dT})$. We prove the theorem by reducing a carefully selected problem to a multi-arm bandit problem with d arms and using the following lower bound for it.

Theorem 1 (Auer et al., 2002b)**.** *For any number of actions $K \geq 2$ and for any time horizon T, there exists a distribution over the assignment of rewards such that the expected regret of any algorithm (where the expectation is taken with respect to both the randomization over rewards and the algorithms internal randomization) is at least*

$$\frac{1}{20} \min\left\{\sqrt{KT}, T\right\}.$$

We now state a lower bound for spectral bandits, featuring the effective dimension d.

Theorem 2. *For any T and d, there exists a problem with effective dimension d and time horizon T such that the expected regret of any algorithm is of $\Omega(\sqrt{dT})$.*

Proof. We define a problem with the set of actions consisting of $K \approx d$ blocks. Each block is a complete graph K_{M_T} on M_T vertices. Moreover, all weights of the edges inside a component are equal to one. We define M_T as a T-dependent constant such that the effective dimension of the problem d is exactly K. We specify the precise value of M_T later.

On top of the structure described above, we choose a reward function with smoothness 0, i.e., a constant on each of the components of the graph. In fact, even knowing that the reward function is constant on individual components, this problem is as difficult as the multi-arm bandit problem with K arms. Therefore, the lower bound of $\Omega(\sqrt{KT})$ of the K-arm bandit problem applies to our setting too. Consequently, we have the lower bound of $\Omega(\sqrt{dT})$, since $d = K$.

The last part of the proof is to show that $d = K$ and therefore, we have to specify value of M_T. The graph consists of K blocks consisting M_T vertices each. Therefore, the graph Laplacian is the following matrix

$$L = \begin{pmatrix} \begin{matrix} (M_T-1) & -1 & \dots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & (M_T-1) \end{matrix} & & 0 \\ & \ddots & \\ 0 & & \begin{matrix} (M_T-1) & -1 & \dots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & (M_T-1) \end{matrix} \end{pmatrix}.$$

Now we can compute eigenvalues of L to obtain

$$L \xrightarrow{\text{eigenvalues}} \overbrace{0, \dots, 0}^{K}, \overbrace{M_T, \dots, M_T}^{(M_T-1)K}.$$

The next step is to plug these eigenvalues to the definition of the effective dimension and to set the value of M_T to obtain $d = K$.

$$d = \left\lceil \frac{\max \log \prod_{i=1}^{K} \left(1 + \frac{t_i}{\lambda}\right) \prod_{i=K+1}^{KM_T} \left(1 + \frac{t_i}{\lambda + M_T}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil$$

Setting $M_T \geq T/K$ we have that the maximum in the definition is obtained for $t_1 = \dots = t_K = T/K$ and $t_{K+1} = \dots = t_{KM_T} = 0$. Therefore,

$$d = \left\lceil \frac{\log \prod_{i=1}^{K} \left(1 + \frac{T}{K\lambda}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = \left\lceil \frac{\sum_{i=1}^{K} \log \left(1 + \frac{T}{K\lambda}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = K$$

This means that our problem is at least as difficult as the multi-arm bandit problem with $K = d$ arms and therefore, the standard lower bound for K-arm bandits (Theorem 1) applies to our problem. □

4 Algorithms

In this section we introduce three algorithms for spectral setting; SpectralUCB, SpectralTS, and SpectralEliminator. For each algorithm, we present a regret bound and later in this section, we discuss computational advantages and compare theoretical regret bounds of the algorithms with the lower bound provided in the previous section. Complete proofs for the regret bounds are provided later in Section 5.

4.1 SpectralUCB algorithm and theoretical guarantees

The first algorithm we present is SpectralUCB (Algorithm 1) which is based on LinUCB [Li et al., 2010] and uses the *spectral penalty* (2.1). Here we consider the regularized least-squares estimate $\hat{\boldsymbol{\alpha}}_t$ of the form:

$$\hat{\boldsymbol{\alpha}}_t = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^N} \left(\sum_{s=1}^{t} \left[\mathbf{x}_{a_s}^\intercal \boldsymbol{w} - r_{a_s} \right]^2 + \|\boldsymbol{w}\|_{\boldsymbol{\Lambda}}^2 \right)$$

A key part of the algorithm is to define the $c_t\|\mathbf{x}\|_{\mathbf{V}_t^{-1}}$ confidence widths for the prediction of the rewards. We take advantage of our analysis (Section 5.3) to define c_t based on the effective dimension d which is specifically tailored to our setting. By doing this we also avoid the computation of the determinant (see Section 5). The following theorem characterizes the performance of SpectralUCB and bounds the regret as a function of effective dimension d.

Theorem 3. *Let d be the effective dimension and λ be the minimum eigenvalue of $\boldsymbol{\Lambda}$. If $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$ and for all $\mathbf{x}_a$, $\mathbf{x}_a^\intercal\boldsymbol{\alpha} \in [-1, 1]$, then the cumulative regret of* SpectralUCB *is with probability at least $1-\delta$ bounded as*

$$R_T \leq \left(2R\sqrt{2d\log(1+T/(K\lambda)) + 8\log(1/\delta)} + 2C + 2 \right) \sqrt{2dT\log(1+T/(K\lambda))}.$$

Remark 3. *The constant C needs to be such that $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$. If we set C too small, the true $\boldsymbol{\alpha}$ will lie outside of the region and far from $\hat{\boldsymbol{\alpha}}_t$, causing the algorithm to underperform. Alternatively, C can be time dependent, e.g., $C_t = \log T$. In such case, we do not need to know an upper bound on $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$ in advance, but our regret bound would only hold after some t, when $C_t \geq \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$.*

Algorithm 1 SpectralUCB

Input:
  N: the number of actions, T: the number of rounds
  $\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$: spectral basis of graph Laplacian $\mathcal{L}$
  λ, δ: regularization and confidence parameters
  R, C: upper bounds on the noise and $\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}$
Initialization:
  $\boldsymbol{V}_1 = \mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$
  $\hat{\boldsymbol{\alpha}}_1 = 0_N$
  $d = \lceil (\max \log \prod_{i=1}^{N} (1 + t_i/\lambda_i)) / \log(1 + T/(K\lambda)) \rceil$ (Definition 1)
  $c = R\sqrt{2d \log(1 + T/(K\lambda)) + 8 \log(1/\delta)} + C$
Run:
for $t = 1$ **to** T **do**
  Choose the node a_t (a_t-th row of $\mathbf{Q}$) such that:
    $a_t = \arg\max_a \left(\mathbf{x}_a^{\mathsf{T}} \hat{\boldsymbol{\alpha}}_t + c \|\mathbf{x}_a\|_{\boldsymbol{V}_t^{-1}} \right)$
  Observe a noisy reward $r_t = \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} + \varepsilon_t$
  Update the basis coefficients $\hat{\boldsymbol{\alpha}}$:
    $\boldsymbol{V}_{t+1} = \boldsymbol{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^{\mathsf{T}}$
    $\hat{\boldsymbol{\alpha}}_{t+1} = \boldsymbol{V}_{t+1}^{-1} \sum_{s=1}^{t} \mathbf{x}_{a_s} r_s$
end for

We provide the proof of Theorem 3 in Section 5 and examine the performance of SpectralUCB experimentally in Section 6. The $d\sqrt{T}$ result of Theorem 3 is to be compared with the classical linear bandits, where LinUCB is the algorithm often used in practice [Li et al., 2010] achieving $D\sqrt{T}$ cumulative regret. As mentioned above and demonstrated in Figures 2.2 and 2.3, in the $T < N$ regime we can expect $d \ll D = N$ and obtain an improved performance.

4.2 SpectralTS algorithm and theoretical guarantees

The second algorithm presented in this paper is SpectralTS which is based on LinearTS [Agrawal and Goyal, 2013] and uses Thompson Sampling to decide which arm to play. Specifically, we represent our current knowledge about $\boldsymbol{\alpha}$ as a normal distribution $\mathcal{N}(\hat{\boldsymbol{\alpha}}_t, v^2 \boldsymbol{V}_t^{-1})$, where $\hat{\boldsymbol{\alpha}}_t$ is our actual approximation of the unknown vector $\boldsymbol{\alpha}$ and $v^2 \boldsymbol{V}_t^{-1}$ reflects our uncertainty about it. As mentioned before, we assume that the reward function is a linear combination of eigenvectors of graph Laplacian $\mathcal{L}$ with large coefficients corresponding to the eigenvectors with small eigenvalues. We encode

Algorithm 2 SpectralTS

Input:
  N: the number of actions, T: the number of rounds
  $\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$: spectral basis of graph Laplacian $\mathcal{L}$
  λ, δ: regularization and confidence parameters
  R, C: upper bounds on the noise and $\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}$
Initialization:
  $\boldsymbol{V}_1 = \mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}_N$
  $\hat{\boldsymbol{\alpha}}_1 = 0_N$
  $d = \lceil (\max \log \prod_{i=1}^{N} (1 + t_i/\lambda_i)) / \log(1 + T/(K\lambda)) \rceil$ (Definition 1)
  $v = R\sqrt{3d \log(1/\delta + T/(\delta \lambda K))} + C$
Run:
for $t = 1$ **to** T **do**
  Sample $\tilde{\boldsymbol{\alpha}}_t \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}_t, v^2 \boldsymbol{V}_t^{-1})$
  Choose the node a_t (a_t-th row of $\mathbf{Q}$):
    $a_t = \arg\max_a \mathbf{x}_a^{\mathsf{T}} \tilde{\boldsymbol{\alpha}}$
  Observe a noisy reward $r_t = \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} + \varepsilon_t$
  Update the basis coefficients $\hat{\boldsymbol{\alpha}}$:
    $\boldsymbol{V}_{t+1} = \boldsymbol{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^{\mathsf{T}}$
    $\hat{\boldsymbol{\alpha}}_{t+1} = \boldsymbol{V}_{t+1}^{-1} \sum_{s=1}^{t} \mathbf{x}_{a_s} r_s$
end for

this assumption into our initial confidence ellipsoid by setting $\boldsymbol{V}_1 = \mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$, where λ is a regularization parameter.

In every time step t we generate a sample $\tilde{\boldsymbol{\alpha}}_t$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\alpha}}_t, v^2 \boldsymbol{V}_t^{-1})$, choose an arm a_t which maximizes $\mathbf{x}_i^{\mathsf{T}} \tilde{\boldsymbol{\alpha}}_t$, and receive a reward. Afterwards, we update our estimate of $\boldsymbol{\alpha}$ and the confidence of it, i.e., we compute $\hat{\boldsymbol{\alpha}}_{t+1}$ and $\boldsymbol{V}_{t+1}$,

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^{\mathsf{T}} \qquad \hat{\boldsymbol{\alpha}}_{t+1} = \boldsymbol{V}_{t+1}^{-1} \left(\sum_{s=1}^{t} \mathbf{x}_{a_s} r_s \right).$$

Remark 4. *Since TS is a Bayesian approach, it requires a prior to run and we choose it here to be a Gaussian. However, this does not pose any assumption whatsoever about the actual data both for the algorithm and the analysis. The only assumptions we make about the data are: (a) that the mean payoff is linear in the features, (b) that the noise is R-sub-Gaussian, and (c) that we know a bound on the Laplacian*

norm of the mean reward function. We provide a frequentist bound on the regret (and not an average over the prior) which is a much stronger worst case result.

The following theorem upper bounds the cumulative regret of SpectralTS in terms of effective dimension.

Theorem 4. *Let d be the effective dimension and λ be the minimum eigenvalue of $\mathbf{\Lambda}$. If $\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}} \leq C$ and for all $\mathbf{x}_a$, $\mathbf{x}_a^\intercal \boldsymbol{\alpha} \in [-1, 1]$, then the cumulative regret of* SpectralTS *is with probability at least $1 - \delta$ bounded as*

$$R_T \leq \frac{11g}{p}\sqrt{\frac{2+2\lambda}{\lambda} dT \log\left(1 + \frac{T}{K\lambda}\right)} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T\log\frac{2}{\delta}},$$

where $p = 1/(4e\sqrt{\pi})$ and

$$g = \sqrt{4\log(TN)}\left(R\sqrt{3d\log\left(\frac{1}{\delta} + \frac{T}{\delta\lambda K}\right)} + C\right) + R\sqrt{d\log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C.$$

Remark 5. *Substituting g and p we see that the regret bound scales as $d\sqrt{T\log N}$. Note that $N = D$ could be exponential in d and we need to consider factor $\sqrt{\log N}$ in our bound. On the other hand, if N is indeed exponential in d, then our algorithm scales with $\log D\sqrt{T\log D} = \log(D)^{3/2}\sqrt{T}$ which is even better.*

4.3 SpectralEliminator algorithm and theoretical guarantees

It is known that the available upper bound for LinUCB, LinearTS or OFUL is not optimal for the linear bandit setting with a finite number of arms in terms of dimension D. On the other hand, the algorithms SupLinRel or SupLinUCB achieve the optimal $\sqrt{DT}$ regret. In the following, we likewise provide an algorithm that also scales better with d and achieves $\sqrt{dT}$ regret. The algorithm is called SpectralEliminator (Algorithm 3) and works in phases, eliminating the arms that are not promising. The phases are defined by the time indexes $t_1 = 1 \leq t_2 \leq \ldots$ and depend on some parameter β. The algorithm is in a spirit similar to the Improved UCB by Auer and Ortner [2010]. In the following theorem we characterize the performance of SpectralEliminator and show that the upper bound on regret has $\sqrt{d}$ improvement over SpectralUCB and SpectralTS.

Algorithm 3 SpectralEliminator

Input:
N : the number of nodes, T : the number of pulls
$\{\mathbf{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$ spectral basis of $\mathcal{L}$
λ : regularization parameter
$\beta, \{t_j\}_j^J$ parameters of the elimination and phases
$A_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.
for $j = 1$ **to** J **do**
  $\boldsymbol{V}_{t_j} = \gamma \mathbf{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$
  for $t = t_j$ **to** $\min(t_{j+1} - 1, T)$ **do**
    Play available arm a_t ($\mathbf{x}_{a_t} \in A_j$) with the largest width and observe r_t:
      $a_t = \arg\max_{a | \mathbf{x}_a \in A_j} \|\mathbf{x}_a\|_{\boldsymbol{V}_t^{-1}}$
    $\boldsymbol{V}_{t+1} = \boldsymbol{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^\intercal$
  end for
  Eliminate the arms that are not promising:
    $\hat{\boldsymbol{\alpha}}_{j+1} = \boldsymbol{V}_{t+1}^{-1} [\mathbf{x}_{t_j}, \ldots, \mathbf{x}_t][r_{t_j}, \ldots, r_t]^\intercal$
    $p = \max_{\mathbf{x} \in A_j} \left[\langle \hat{\boldsymbol{\alpha}}_{j+1}, \mathbf{x} \rangle - \|\mathbf{x}\|_{\boldsymbol{V}_{t+1}^{-1}} \beta \right]$
    $A_{j+1} = \left\{ \mathbf{x} \in A_j, \langle \hat{\boldsymbol{\alpha}}_{j+1}, \mathbf{x} \rangle + \|\mathbf{x}\|_{\boldsymbol{V}_{t+1}^{-1}} \beta \geq p \right\}$
end for

Theorem 5. *Choose the phases starts as* $t_j = 2^{j-1}$. *Assume all rewards are in* $[0, 1]$ *and* $\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}} \leq C$. *For any* $\delta > 0$, *with probability at least* $1 - \delta$, *the cumulative regret of* SpectralEliminator *algorithm run with parameter* $\beta = R\sqrt{\log(2K(1 + \log_2 T)/\delta)} + C$ *is bounded as:*

$$R_T \leq 2 + 8 \left(R\sqrt{2 \log \frac{2K(1 + \log_2 T)}{\delta}} + C + \frac{1}{2} \right) \sqrt{2dT \log_2(T) \log\left(1 + T/(\lambda K)\right)}$$

4.4 Scalability and computational complexity

There are three main computational issues to address in order to make proposed algorithms scalable: the computation of N UCBs (apply to SpectralUCB), matrix inversion, and obtaining the eigenbasis which serves as an input to the algorithm. First, to speed up the computation of N UCBs (in general done in N^3 time) in each time step, we use lazy updates technique [Desautels et al., 2012] which maintains a

sorted queue of UCBs and using the fact that UCB for every arm can only decrease after an update. Therefore, the algorithm does not need to update all UCBs in each time step. This in practice leads to substantial speed gains. This issue does not apply to SpectralTS since we only need to sample $\tilde{\boldsymbol{\alpha}}$ which can be done in N^2 time and find a maximum of $\mathbf{x}_i^\intercal \tilde{\boldsymbol{\alpha}}$ which can be also done in N^2 time. In general, the computational complexity of sampling in SpectralTS is better than the complexity of computing N UCBs in SpectralUCB. However, using lazy updates can significantly speed up SpectralUCB up to the point that SpectralUCB can be comparable to the SpectralTS.

Second, all of the proposed algorithms need to compute inverse if $N \times N$ matrix in each time step which can be costly. However, we can use Sherman-Morrison formula to invert matrix iteratively and thus speed up matrix inversion since the matrix changes only by adding a rank-1 matrix from one time step to the next one.

Finally, while the eigendecomposition of a general matrix is computationally difficult, Laplacians are symmetric diagonally dominant (SDD). This enables us to use fast SDD solvers as CMG by Koutis et al. [2011]. Furthermore, using CMG we can find good approximations to the first L eigenvectors in $\mathcal{O}(Lm \log m)$ time, where m is the number of edges in the graph (e.g. $m = 10N$ in the Flixter experiment). CMG can easily work with N in millions. In general, we have $L = N$ but from our experience, a smooth reward function can be often approximated by dozens of eigenvectors. In fact, L can be considered as an upper bound on the number of eigenvectors we actually need. Furthermore, by choosing small L we not only reduce the complexity of eigendecomposition but also the complexity of the least-square problem being solved in each iteration.

Choosing a small L can significantly reduce the computation but it is important to choose L large enough so that still less than L eigenvectors are enough. This way, the problem that we solve is still relevant and our analysis applies. In short, the problem cannot be solved trivially by choosing first k relevant eigenvectors because k is unknown. Therefore, in practice, we choose the largest L such that our method is able to run. In Section 6.3, we demonstrate that we can obtain good results with relatively small L.

5 Analysis

Now we are ready to prove regret bounds for individual algorithms. First, we show some general preliminary results in Section 5.1. Then we present several auxiliary lemmas concerning confidence ellipsoid (Section 5.2) and effective dimension (Section 5.3). Using these results we upper-bound the regrets of SpectralUCB (Section 5.4), SpectralTS (Section 5.5), and SpectralEliminator (Section 5.6).

5.1 Preliminaries

Lemma 2. *For a Gaussian distributed random variable Z with mean m and variance σ^2, for any $z \geq 1$,*

$$\frac{1}{2\sqrt{\pi}z} e^{-z^2/2} \leq \mathbb{P}(|Z - m| > \sigma z) \leq \frac{1}{\sqrt{\pi}z} e^{-z^2/2}.$$

Multiple use of Sylvester's determinant theorem gives:

Lemma 3. *Let $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$, then*

$$\log \frac{|\boldsymbol{V}_t|}{|\boldsymbol{\Lambda}|} = \sum_{s=1}^{t-1} \log(1 + \|\mathbf{x}_s\|^2_{\boldsymbol{V}_s^{-1}})$$

Lemma 4. *For any symmetric, positive semi-definite matrix $\boldsymbol{X}$ and any vectors $\mathbf{u}$ and $\mathbf{y}$:*

$$\mathbf{y}^\intercal (\boldsymbol{X} + \mathbf{u}\mathbf{u}^\intercal)^{-1} \mathbf{y} \leq \mathbf{y}^\intercal \boldsymbol{X}^{-1} \mathbf{y}$$

Proof. Using Sherman–Morrison formula and the fact that inverse of a symmetric

matrix is symmetric again, we have

$$\begin{aligned} -\frac{\left(\mathbf{u}^\intercal \boldsymbol{X}^{-1}\mathbf{y}\right)^\intercal \left(\mathbf{u}^\intercal \boldsymbol{X}^{-1}\mathbf{y}\right)}{1+\mathbf{u}^\intercal \boldsymbol{X}^{-1}\mathbf{u}} &\leq 0 \\ \mathbf{y}^\intercal \left(\boldsymbol{X}^{-1} - \frac{\boldsymbol{X}^{-1}\mathbf{u}\mathbf{u}^\intercal \boldsymbol{X}^{-1}}{1+\mathbf{u}^\intercal \boldsymbol{X}^{-1}\mathbf{u}}\right)\mathbf{y} &\leq \mathbf{y}^\intercal \boldsymbol{X}^{-1}\mathbf{y} \\ \mathbf{y}^\intercal \left(\boldsymbol{X}+\mathbf{u}\mathbf{u}^\intercal\right)^{-1}\mathbf{y} &\leq \mathbf{y}^\intercal \boldsymbol{X}^{-1}\mathbf{y}. \end{aligned}$$

□

Corollary 1. *Let* $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$. *Then for any vector* $\mathbf{x}$

$$||\mathbf{x}||_{\boldsymbol{V}_{t_1}^{-1}} \geq ||\mathbf{x}||_{\boldsymbol{V}_{t_2}^{-1}}$$

holds for any positive integers t_1, t_2 *satisfying* $t_1 \leq t_2$.

5.2 Confidence ellipsoid

The first two lemmas are by Abbasi-Yadkori et al. [2011] and we restate them for convenience.

Lemma 5. *Let* $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$ *and define* $\boldsymbol{\xi}_t = \sum_{s=1}^{t-1} \varepsilon_s \mathbf{x}_s$. *With probability at least* $1-\delta$, $\forall t \geq 1$*:*

$$\|\boldsymbol{\xi}_t\|^2_{\boldsymbol{V}_t^{-1}} \leq 2R^2 \log\left(\frac{|\boldsymbol{V}_t|^{1/2}}{\delta |\boldsymbol{\Lambda}|^{1/2}}\right)$$

Lemma 6. *For any* t, *let* $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$. *Then:*

$$\sum_{s=1}^{t} \min\left(1,\ \|\mathbf{x}_s\|^2_{\boldsymbol{V}_s^{-1}}\right) \leq 2 \log \frac{|\boldsymbol{V}_{t+1}|}{|\boldsymbol{\Lambda}|}$$

The next lemma is a generalization of Theorem 2 by Abbasi-Yadkori et al. [2011] to the regularization with $\boldsymbol{\Lambda}$.

Lemma 7. *Let* $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$ *and* $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$*. With probability at least* $1-\delta$*, for any vector* $\mathbf{x}$ *and for any positive integer* t*:*

$$|\mathbf{x}^\intercal \hat{\boldsymbol{\alpha}}_t - \mathbf{x}^\intercal \boldsymbol{\alpha}| \leq \|\mathbf{x}\|_{\boldsymbol{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\boldsymbol{V}_t|^{1/2}}{\delta |\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right)$$

Proof. We have:

$$\begin{aligned}
|\mathbf{x}^\intercal \hat{\boldsymbol{\alpha}}_t - \mathbf{x}^\intercal \boldsymbol{\alpha}| &= |\mathbf{x}^\intercal(-\boldsymbol{V}_t^{-1}\boldsymbol{\Lambda}\boldsymbol{\alpha} + \boldsymbol{V}_t^{-1}\boldsymbol{\xi}_t)| \\
&\leq |\mathbf{x}^\intercal \boldsymbol{V}_t^{-1}\boldsymbol{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^\intercal \boldsymbol{V}_t^{-1}\boldsymbol{\xi}_t| \\
&\leq |\mathbf{x}^\intercal \boldsymbol{V}_t^{-\frac{1}{2}} \boldsymbol{V}_t^{-\frac{1}{2}}\boldsymbol{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^\intercal \boldsymbol{V}_t^{-\frac{1}{2}} \boldsymbol{V}_t^{-\frac{1}{2}}\boldsymbol{\xi}_t| \\
&\leq \|\mathbf{x}\|_{\boldsymbol{V}_t^{-1}} \left(\|\boldsymbol{\xi}_t\|_{\boldsymbol{V}_t^{-1}} + \|\boldsymbol{\Lambda}\boldsymbol{\alpha}\|_{\boldsymbol{V}_t^{-1}} \right),
\end{aligned}$$

where we used Cauchy-Schwarz inequality in the last step. Now we bound $\|\boldsymbol{\xi}_t\|_{\boldsymbol{V}_t^{-1}}$ by Lemma 5 and using Corollary 1 we bound $||\boldsymbol{\Lambda}\boldsymbol{\alpha}||_{\boldsymbol{V}_t^{-1}}$ as

$$\|\boldsymbol{\Lambda}\boldsymbol{\alpha}\|_{\boldsymbol{V}_t^{-1}} \leq \|\boldsymbol{\Lambda}\boldsymbol{\alpha}\|_{\boldsymbol{V}_1^{-1}} = \|\boldsymbol{\Lambda}\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}^{-1}} = \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C.$$

□

5.3 Effective dimension

In Section 5.2 we showed that several quantities scale with $\log(|\boldsymbol{V}_t|/|\boldsymbol{\Lambda}|)$, which can be of order D. Therefore, in this part, we present the key ingredient of our analysis, based on the geometrical properties of determinants (Lemmas 9 and 10), to upper-bound $\log(|\boldsymbol{V}_t|/|\boldsymbol{\Lambda}|)$ by a term that scales with d (Lemma 11). Not only this will allow us to show that the regret bound scales with d, but it also helps us to avoid the computation of the determinants in Algorithm 1.

Lemma 8. *For any real positive-definite matrix* A *with only simple eigenvalue multiplicities and any vector* $\mathbf{x}$ *such that* $\|\mathbf{x}\|_2 \leq 1$ *we have that the determinant* $|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal|$ *is maximized by a vector* $\mathbf{x}$ *which is aligned with an eigenvector of* $\mathbf{A}$*.*

Proof. Using Sylvester's determinant theorem, we have:

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\intercal| = |\mathbf{A}|(1 + \mathbf{x}^\intercal \mathbf{A}^{-1}\mathbf{x})$$

From the spectral theorem, there exists an orthonormal matrix $\mathbf{U}$, the columns of which are the eigenvectors of $\mathbf{A}$; such that $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\intercal$ with $\mathbf{D}$ being a diagonal matrix with the positive eigenvalues of $\mathbf{A}$ on the diagonal. Thus:

$$\max_{\|\mathbf{x}\|_2 \leq 1} \mathbf{x}^\intercal \mathbf{A}^{-1} \mathbf{x} = \max_{\|\mathbf{x}\|_2 \leq 1} \mathbf{x}^\intercal \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^\intercal \mathbf{x} = \max_{\|\mathbf{y}\|_2 \leq 1} \mathbf{y}^\intercal \mathbf{D}^{-1} \mathbf{y},$$

since $\mathbf{U}$ is a bijection from $\{\mathbf{x}, \|\mathbf{x}\|_2 \leq 1\}$ to itself.

Since there are no multiplicities, it is easy to see that the quadratic mapping $\mathbf{y} \mapsto \mathbf{y}^\intercal \mathbf{D}^{-1}\mathbf{y}$ is maximized (under the constraint $\|\mathbf{y}\|_2 \leq 1$) by a canonical vector $\boldsymbol{e}_I$ corresponding to the lowest diagonal entry I of $\mathbf{D}$. Thus the maximum of $\mathbf{x} \mapsto \mathbf{x}^\intercal \mathbf{A}^{-1}\mathbf{x}$ is reached for $\mathbf{U}\boldsymbol{e}_I$, which is the eigenvector of $\mathbf{A}$ corresponding to its lowest eigenvalue. □

Lemma 9. *Let $\boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_N)$ be any diagonal matrix with strictly positive entries. For any vectors $(\mathbf{x}_s)_{1 \leq s < t}$ such that $\|\mathbf{x}_s\|_2 \leq 1$ for all $1 \leq s < t$, we have that the determinant $|\boldsymbol{V}_t|$ of $\boldsymbol{V}_t = \boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$ is maximized when all $\mathbf{x}_s$ are aligned with the axes.*

Proof. Let us write $d(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) = |\boldsymbol{V}_t|$ the determinant of $\boldsymbol{V}_t$. We want to characterize:

$$\max_{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1} : \|\mathbf{x}_s\|_2 \leq 1, \forall 1 \leq s < t} d(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$$

For any $1 \leq i < t$, let us define:

$$\boldsymbol{V}_{-i} = \boldsymbol{\Lambda} + \sum_{\substack{s=1 \\ s \neq i}}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$$

We have that $\boldsymbol{V}_t = \boldsymbol{V}_{-i} + \mathbf{x}_i \mathbf{x}_i^\intercal$. Consider the case where every eigenvalue is of multiplicities one. In this case, Lemma 8 implies that $\mathbf{x}_i \mapsto d(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{t-1})$ is maximized when $\mathbf{x}_i$ is aligned with an eigenvector of $\boldsymbol{V}_{-i}$. Thus all $\mathbf{x}_s$, for $1 \leq s < t$, are aligned with an eigenvector of $\boldsymbol{V}_{-i}$ and therefore also with an eigenvector of $\boldsymbol{V}_t$. Consequently, the eigenvectors of $\sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$ are also aligned with $\boldsymbol{V}_t$. Since $\boldsymbol{\Lambda} = \boldsymbol{V}_t - \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\intercal$ and $\boldsymbol{\Lambda}$ is diagonal, we conclude that $\boldsymbol{V}_t$ is diagonal and all $\mathbf{x}_s$ are aligned with the canonical axes.

Now in the case of eigenvalue multiplicities, the maximum of $|\boldsymbol{V}_t|$ may be reached by several sets of vectors $\{(\mathbf{x}_s^m)_{1\leq s<t}\}_m$ but for some m^*, the set $(\mathbf{x}_s^{m^*})_{1\leq s<t}$ will be aligned with the axes. In order to see that, consider a perturbed matrix $\boldsymbol{V}_{-i}^\varepsilon$ by a random perturbation of amplitude at most ε, i.e. such that $\boldsymbol{V}_{-i}^\varepsilon \to \boldsymbol{V}_{-i}$ when $\varepsilon \to 0$. Since the perturbation is random, then the probability that $\boldsymbol{\Lambda}^\varepsilon$, as well as all other $\boldsymbol{V}_{-i}^\varepsilon$ possess an eigenvalue of multiplicity bigger than 1 is zero. Since the mapping $\varepsilon \mapsto \boldsymbol{V}_{-i}^\varepsilon$ is continuous, we deduce that any adherent point $\bar{\mathbf{x}}_i$ of the sequence $(\mathbf{x}_i^\varepsilon)_\varepsilon$ (there exists at least one since the sequence is bounded in ℓ_2-norm) is aligned with the limit $\boldsymbol{V}_{-i}$ and we can apply the previous reasoning. □

Lemma 10. *For any t, let $\boldsymbol{V}_t = \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\intercal + \boldsymbol{\Lambda}$. Then:*

$$\log \frac{|\boldsymbol{V}_t|}{|\boldsymbol{\Lambda}|} \leq \max \sum_{i=1}^{N} \log\Big(1 + \frac{t_i}{\lambda_i}\Big),$$

where the maximum is taken over all possible positive real numbers $\{t_1, \ldots, t_N\}$, such that $\sum_{i=1}^N t_i = t-1$.

Proof. We want to bound the determinant $|\boldsymbol{V}_t|$ under the coordinate constraints $\|\mathbf{x}_s\|_2 \leq 1$. Let:

$$M(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) = \Big|\boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\intercal\Big|$$

From Lemma 9 we deduce that the maximum of M is reached when all $\mathbf{x}_t$ are aligned with the axes:

$$\begin{aligned} M &= \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1};\mathbf{x}_s\in\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_N\}} \Big|\boldsymbol{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\intercal\Big| \\ &= \max_{t_1,\ldots,t_N \text{ positive integers}, \sum_{i=1}^N t_i = t-1} \Big|\operatorname{diag}(\lambda_i + t_i)\Big| \\ &\leq \max_{t_1,\ldots,t_N \text{ positive reals}, \sum_{i=1}^N t_i = t-1} \prod_{i=1}^{N} \Big(\lambda_i + t_i\Big), \end{aligned}$$

from which we obtain the result. □

Lemma 11. *Let d be the effective dimension and $t \leq T+1$. Then:*

$$\log \frac{|\boldsymbol{V}_t|}{|\boldsymbol{\Lambda}|} \leq d \log \left(1 + \frac{T}{K\lambda}\right)$$

Proof. Using Lemma 10 and Definition 1 we have:

$$\begin{aligned}
\log \frac{|\boldsymbol{V}_t|}{|\boldsymbol{\Lambda}|} &\leq \max \sum_{i=1}^{N} \log \left(1 + \frac{t_i}{\lambda_i}\right) \\
&= \frac{\max \sum_{i=1}^{N} \log \left(1 + \frac{t_i}{\lambda_i}\right)}{\log(1 + T/(K\lambda))} \log \left(1 + \frac{T}{K\lambda}\right) \\
&\leq \left\lceil \frac{\max \sum_{i=1}^{N} \log \left(1 + \frac{t_i}{\lambda_i}\right)}{\log(1 + T/(K\lambda))} \right\rceil \log \left(1 + \frac{T}{K\lambda}\right) \\
&= d \log \left(1 + \frac{T}{K\lambda}\right).
\end{aligned}$$

□

5.4 Regret bound of SpectralUCB

The analysis of SpectralUCB has two, previously mentioned, main ingredients. The first one is the derivation of the confidence ellipsoid for $\hat{\boldsymbol{\alpha}}$, which is a straightforward update of the analysis of OFUL [Abbasi-Yadkori et al., 2011] using self-normalized martingale inequality (Section 5.2). The second part is crucial to prove that the final regret bound scales only with the effective dimension d and not with the ambient dimension D. We achieve this by considering the geometrical properties of the determinant which holds in our setting (Section 5.3).

Proof of Theorem 3. Let $\mathbf{x}_* = \arg\max_{\mathbf{x}_v} \mathbf{x}_v^\intercal \boldsymbol{\alpha}$ and let $R_T(t)$ denote the instantaneous

regret at time t. With probability at least $1-\delta$, for all t:

$$\begin{aligned} R_T(t) &= \mathbf{x}_*^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} \\ &\leq \mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} - \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} && (2.2) \\ &\leq \mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} - \mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} && (2.3) \\ &= 2c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}. \end{aligned}$$

Inequality (2.2) is by the algorithm design and reflects the optimistic principle of SpectralUCB. Specifically, $\mathbf{x}_*^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_*\|_{\boldsymbol{V}_t^{-1}} \leq \mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}$, from which:

$$\mathbf{x}_*^\mathsf{T}\boldsymbol{\alpha} \leq \mathbf{x}_*^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_*\|_{\boldsymbol{V}_t^{-1}} \leq \mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}$$

In (2.3) we applied Lemma 7: $\mathbf{x}_{a_t}^\mathsf{T}\hat{\boldsymbol{\alpha}}_t \leq \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} + c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}$. Finally, by Lemmas 6 and 11:

$$\begin{aligned} R_T &= \sum_{t=1}^{T} R_T(t) \leq \sum_{t=1}^{T} \min\left(2,\, 2c\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}\right) \leq (2+2c)\sum_{t=1}^{T} \min\left(1,\, \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}\right) \\ &\leq (2+2c)\sqrt{T\sum_{t=1}^{T} \min\left(1,\, \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right)} \leq (2+2c)\sqrt{2T\log\frac{|\boldsymbol{V}_{T+1}|}{|\boldsymbol{\Lambda}|}} \\ &\leq (2+2c)\sqrt{2dT\log\left(1+\frac{T}{K\lambda}\right)} \end{aligned}$$

By plugging c, we get that with probability at least $1-\delta$, the theorem holds. □

Remark 6. *Notice that if we set $\boldsymbol{\Lambda} = \mathbf{I}$ in Algorithm 1, we recover LinUCB. Since $\log(|\boldsymbol{V}_{T+1}|/|\boldsymbol{\Lambda}|)$ can be upperbounded by $D\log T$ [Abbasi-Yadkori et al., 2011], we obtain $\tilde{\mathcal{O}}(D\sqrt{T})$ upper bound of regret of* LinUCB *as a corollary of Theorem 3.*

5.5 Regret bound of SpectralTS

Regret bound of SpectralTS algorithm is based on the proof technique of Agrawal and Goyal [2013]. The summary of the technique follows. Each time an arm is played, our algorithm improves the confidence about our actual estimate of $\boldsymbol{\alpha}$ via

the update of $\boldsymbol{V}_t$ and thus the update of confidence ellipsoid. However, when we play a suboptimal arm, the regret we obtain can be much higher than the improvement of our knowledge. To overcome this difficulty, the arms are divided into two groups of *saturated* and *unsaturated* arms, based on whether the standard deviation for an arm is smaller than the standard deviation of the optimal arm (Definition 4) or not. Consequently, the optimal arm is in the group of unsaturated arms. The idea is to bound the regret of playing an unsaturated arm in terms of standard deviation and to show that the probability that the saturated arm is played is small enough. This way we overcome the difficulty of high regret and small knowledge obtained by playing an arm.

Definition 3. *We define $E^{\hat{\boldsymbol{\alpha}}}(t)$ as the event that for all i,*

$$|\mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\alpha}}_t - \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}| \leq l\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}$$

where

$$l = R\sqrt{d\log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C,$$

and $E^{\tilde{\boldsymbol{\alpha}}}(t)$ as the event that for all i,

$$|\mathbf{x}_i^\mathsf{T}\tilde{\boldsymbol{\alpha}}_t - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\alpha}}_t| \leq v\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}\sqrt{4\log(TN)}$$

where

$$v = R\sqrt{3d\log\left(\frac{1}{\delta} + \frac{T}{\delta\lambda K}\right)} + C.$$

Definition 4. *Let $\Delta_i = \mathbf{x}_{a^*}^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}$. We say that an arm i is **saturated** at time t if $\Delta_i > g\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}$, and **unsaturated** otherwise (including the optimal arm a_*). Let $C(t)$ denote the set of saturated arms at time t.*

Definition 5. *We define filtration $\mathcal{F}_{t-1}$ as the union of the history until time $t-1$ and features, i.e.,*

$$\mathcal{F}_{t-1} = \{\mathcal{H}_{t-1}\} \cup \{\mathbf{x}_i, i = 1, \ldots, N\}$$

By definition, $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_{T-1}$.

Lemma 12. *For all t, $0 < \delta < 1$, $\mathbb{P}(E^{\hat{\boldsymbol{\alpha}}}(t)) \geq 1 - \delta/T^2$ and for all possible filtrations $\mathcal{F}_{t-1}$,*

$$\mathbb{P}(E^{\tilde{\boldsymbol{\alpha}}}(t) \,|\, \mathcal{F}_{t-1}) \geq 1 - 1/T^2.$$

Proof. **Bounding the probability of event $E^{\hat{\boldsymbol{\alpha}}}(t)$:** Using Lemma 7, where C is such that $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$, for all t and for all i with probability at least $1 - \delta'$ we have

$$\begin{aligned}
|\mathbf{x}_i^\intercal(\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})| &\leq \|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}} \left(R\sqrt{2\log\left(\frac{|\boldsymbol{V}_t|^{1/2}}{\delta'|\boldsymbol{\Lambda}|^{1/2}}\right)} + C \right) \\
&= \|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}} \left(R\sqrt{\log\frac{|\boldsymbol{V}_t|}{|\boldsymbol{\Lambda}|} + 2\log\frac{1}{\delta'}} + C \right).
\end{aligned}$$

Therefore, using Lemma 11 and substituting $\delta' = \delta/T^2$, we get that with probability at least $1 - \delta/T^2$, for all i,

$$\begin{aligned}
|\mathbf{x}_i^\intercal(\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})| &\leq \|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}} \left(R\sqrt{d\log\left(1 + \frac{T}{K\lambda}\right) + d\log\frac{T^2}{\delta}} + C \right) \\
&= \|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}} \left(R\sqrt{d\log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C \right) = l\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}.
\end{aligned}$$

Bounding the probability of event $E^{\tilde{\boldsymbol{\alpha}}}(t)$: The probability of each individual term $|\mathbf{x}_i^\intercal(\tilde{\boldsymbol{\alpha}}_t - \hat{\boldsymbol{\alpha}}_t)| < \sqrt{4\log(TN)}$ can be bounded using Lemma 2 to get

$$\mathbb{P}\left(|\mathbf{x}_i^\intercal(\tilde{\boldsymbol{\alpha}}_t - \hat{\boldsymbol{\alpha}}_t)| \geq v\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}\sqrt{4\log(TN)}\right) \leq \frac{e^{-2\log(TN)}}{\sqrt{\pi 4\log(TN)}} \leq \frac{1}{T^2N}.$$

We complete the proof by taking a union bound over all N vectors $\mathbf{x}_i$. Notice that we took a different approach than Agrawal and Goyal [2013] to avoid the dependence on the ambient dimension D. □

Lemma 13. *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{P}\left(\mathbf{x}_{a_*}^\intercal\tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_*}^\intercal\boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) \geq \frac{1}{4e\sqrt{\pi}}.$$

Proof. Given $\mathcal{F}_{t-1}$, $\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t$ is a Gaussian random variable with the mean $\mathbf{x}_{a_*}^\intercal \hat{\boldsymbol{\alpha}}_t$ and the standard deviation $v\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}$, we can use the anti-concentration inequality in Lemma 2,

$$\begin{aligned}\mathbb{P}\left(\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t \geq \mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) &= \mathbb{P}\left(\frac{\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t - \mathbf{x}_{a_*}^\intercal \hat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}} \geq \frac{\mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha} - \mathbf{x}_{a_*}^\intercal \hat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}} \,|\, \mathcal{F}_{t-1}\right) \\ &\geq \frac{1}{4\sqrt{\pi} Z_t} e^{-Z_t^2},\end{aligned}$$

where

$$|Z_t| = \left|\frac{\mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha} - \mathbf{x}_{a_*}^\intercal \hat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}}\right|.$$

Since we consider a filtration $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true, we can upper bound the numerator to get

$$|Z_t| \leq \frac{l\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}}{v\|\mathbf{x}_{a_*}\|_{\boldsymbol{V}_t^{-1}}} = \frac{l}{v} \leq 1.$$

Finally,

$$\mathbb{P}\left(\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) \geq \frac{1}{4e\sqrt{\pi}}.$$

□

Lemma 14. *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{P}(a_t \notin C(t) \,|\, \mathcal{F}_{t-1}) \geq \frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2}.$$

Proof. The algorithm chooses the arm with the highest value of $\mathbf{x}_i^\intercal \tilde{\boldsymbol{\alpha}}_t$ to be played at time t. Therefore if $\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t$ is greater than $\mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t$ for all saturated arms, i.e., $\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t$, $\forall j \in C(t)$, then one of the unsaturated arms (which include the optimal arm and other suboptimal unsaturated arms) must be played. Therefore,

$$\mathbb{P}(a_t \notin\ C(t) \,|\, \mathcal{F}_{t-1}) \geq \mathbb{P}(\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t,\, \forall j \in C(t) \,|\, \mathcal{F}_{t-1}).$$

By definition, for all saturated arms, i.e., for all $j \in C(t)$, $\Delta_j > g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}}$. Now if both of the events $E^{\hat{\boldsymbol{\alpha}}(t)}$ and $E^{\tilde{\boldsymbol{\alpha}}(t)}$ are true then, by definition of these events, for all $j \in C(t)$, $\mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t \leq \mathbf{x}_j^\intercal \boldsymbol{\alpha}_t + g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}}$. Therefore, given the filtration $\mathcal{F}_{t-1}$, such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true, either $E^{\tilde{\boldsymbol{\alpha}}}(t)$ is false, or else for all $j \in C(t)$,

$$\mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t \leq \mathbf{x}_j^\intercal \boldsymbol{\alpha} + g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}} \leq \mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha}.$$

Hence, for any $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true,

$$\begin{aligned}\mathbb{P}(\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\intercal \tilde{\boldsymbol{\alpha}}_t,\, \forall j \in C(t) \,|\, \mathcal{F}_{t-1}) &\geq \mathbb{P}(\mathbf{x}_{a_*}^\intercal \tilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_*}^\intercal \boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}) - \mathbb{P}\left(\overline{E^{\hat{\boldsymbol{\alpha}}}(t)} \,|\, \mathcal{F}_{t-1}\right) \\ &\geq \frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2}.\end{aligned}$$

In the last inequality we used Lemma 12 and Lemma 13. □

Lemma 15. *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{E}[\Delta_{a_t} \,|\, \mathcal{F}_{t-1}] \leq \frac{11g}{p}\mathbb{E}[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}] + \frac{1}{T^2}$$

Proof. Let $\overline{a}_t$ denote the unsaturated arm with the smallest norm $\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$, i.e.,

$$\overline{a}_t = \operatorname*{arg\,min}_{i \notin C(t)} \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}.$$

Notice that given $\mathcal{F}_{t-1}$, $C(t)$ and $\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$ are deterministic for all i. Therefore, $\overline{a}_t$ is deterministic as well. Now, using Lemma 14, for any $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true,

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}] &\geq \mathbb{E}[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}, a_t \notin C(t)]\, \mathbb{P}(a_t \notin C(t) \,|\, \mathcal{F}_{t-1}) \\ &\geq \|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} \left(\frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2}\right).\end{aligned}$$

Now, if the events $E^{\hat{\boldsymbol{\alpha}}}(t)$ and $E^{\tilde{\boldsymbol{\alpha}}}(t)$ are true, then for all i, by definition, $\mathbf{x}_i^\intercal \tilde{\boldsymbol{\alpha}}_t \leq$

$\mathbf{x}_i^\intercal \boldsymbol{\alpha} + g\|\mathbf{x}_i\|_{\boldsymbol{V}_t^{-1}}$. Using this observation along with $\mathbf{x}_{a_t}^\intercal \tilde{\boldsymbol{\alpha}}_t \geq \mathbf{x}_i^\intercal \tilde{\boldsymbol{\alpha}}_t$ for all i,

$$\begin{aligned}
\Delta_{a_t} &= \Delta_{\bar{a}_t} + (\mathbf{x}_{\bar{a}_t}^\intercal \boldsymbol{\alpha} - \mathbf{x}_{a_t}^\intercal \boldsymbol{\alpha}) \\
&\leq \Delta_{\bar{a}_t} + (\mathbf{x}_{\bar{a}_t}^\intercal \tilde{\boldsymbol{\alpha}}_t - \mathbf{x}_{a_t}^\intercal \tilde{\boldsymbol{\alpha}}_t) \\
&\quad + g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \\
&\leq \Delta_{\bar{a}_t} + g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \\
&\leq g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}.
\end{aligned}$$

Therefore, for any $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true, either $\Delta_{a_t} \leq 2g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}$, or $E^{\tilde{\boldsymbol{\alpha}}}(t)$ is false. We can deduce that

$$\begin{aligned}
\mathbb{E}[\Delta_{a_t} \,|\, \mathcal{F}_{t-1}] &\leq \mathbb{E}\left[2g\|\mathbf{x}_{\bar{a}_t}\|_{\boldsymbol{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \mathbb{P}\left(\overline{E^{\tilde{\boldsymbol{\alpha}}}(t)}\right) \\
&\leq \frac{2g}{p - \frac{1}{T^2}} \mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + g\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \frac{1}{T^2} \\
&\leq \frac{11g}{p} \mathbb{E}[\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}] + \frac{1}{T^2}.
\end{aligned}$$

In the last inequality we used that $1/(p - 1/T^2) \leq 5/p$, which holds trivially for $T \leq 4$. For $T \geq 5$, we get that $T^2 \geq 5e\sqrt{\pi}$, which holds for $T \geq 5$. □

Definition 6. *We define* $R'_T(t) = R_T(t) \cdot I(E^{\hat{\boldsymbol{\alpha}}}(t))$.

Definition 7. *A sequence of random variables* $(Y_t; t \geq 0)$ *is called a* ***super-martingale*** *corresponding to a filtration* $\mathcal{F}_t$*, if for all* t*,* Y_t *is* $\mathcal{F}_t$*-measurable, and for* $t \geq 1$*,*

$$\mathbb{E}[Y_t - Y_{t-1} \,|\, \mathcal{F}_{t-1}] \leq 0.$$

Next, following Agrawal and Goyal [2013], we establish a super-martingale process that will form the basis of our proof of the high-probability regret bound.

Definition 8. *Let*

$$\begin{aligned}
X_t &= R'_T(t) - \frac{11g}{p}\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} - \frac{1}{T^2} \\
Y_t &= \sum_{w=1}^{t} X_w.
\end{aligned}$$

Lemma 16. *$(Y_t; t = 0, \dots, T)$ is a super-martingale process with respect to $\mathcal{F}_t$.*

Proof. We need to prove that for all $t \in \{1, \dots, T\}$, and any possible $\mathcal{F}_{t-1}$, $\mathbb{E}[Y_t - Y_{t-1} \,|\, \mathcal{F}_{t-1}] \leq 0$, i.e.

$$\mathbb{E}[R'_T(t) \,|\, \mathcal{F}_{t-1}] \leq \frac{11g}{p} \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} + \frac{1}{T^2}.$$

Note that whether $E^{\hat{\boldsymbol{\alpha}}}(t)$ is true or not, is completely determined by $\mathcal{F}_{t-1}$. If $\mathcal{F}_{t-1}$ is such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ is not true, then $R'_T(t) = R_T(t) \cdot I(E^{\hat{\boldsymbol{\alpha}}}(t)) = 0$, and the above inequality holds trivially. Moreover, for $\mathcal{F}_{t-1}$ such that $E^{\hat{\boldsymbol{\alpha}}}(t)$ holds, the inequality follows from Lemma 15. □

Unlike [Agrawal and Goyal, 2013, Abbasi-Yadkori et al., 2011], we do not want to require $\lambda \geq 1$. Therefore, we provide the following lemma that shows the dependence of $\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}$ on λ.

Lemma 17. *For all t,*

$$\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}} \leq \left(2 + \frac{2}{\lambda}\right) \log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right).$$

Proof. Note, that $\|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \leq (1/\sqrt{\lambda})\|\mathbf{x}_{a_t}\| \leq (1/\sqrt{\lambda})$ and for all $0 \leq x \leq 1$ we have

$$x \leq 2\log(1 + x). \tag{2.4}$$

Now we consider two cases depending on λ. If $\lambda \geq 1$, we know that $0 \leq \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} \leq 1$ and therefore by (2.4),

$$\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}} \leq 2\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right).$$

Similarly, if $\lambda < 1$, then $0 \leq \lambda\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}} \leq 1$ and we get

$$\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}} \leq \frac{2}{\lambda}\log\left(1 + \lambda\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right) \leq \frac{2}{\lambda}\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right).$$

Combining the two, we get that for all $\lambda \geq 0$,

$$\|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}} \leq \max\left(2, \frac{2}{\lambda}\right) \log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right) \leq \left(2 + \frac{2}{\lambda}\right) \log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right).$$

□

Proof of Theorem 4. First, notice that X_t is bounded as $|X_t| \leq 1 + 11g/(p\sqrt{\lambda}) + 1/T^2 \leq (11/\sqrt{\lambda} + 2)g/p$. Thus, we can apply Azuma-Hoeffding inequality to obtain that with probability at least $1 - \delta/2$,

$$\sum_{t=1}^{T} R'_T(t) \leq \sum_{t=1}^{T} \frac{11g}{p} \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} + \sum_{t=1}^{T} \frac{1}{T^2} + \sqrt{2\left(\sum_{t=1}^{T} \frac{g^2}{p^2}\left(\frac{11}{\sqrt{\lambda}} + 2\right)^2\right) \log\frac{2}{\delta}}.$$

Since p and g are constants, then with probability $1 - \delta/2$,

$$\sum_{t=1}^{T} R'_T(t) \leq \frac{11g}{p} \sum_{t=1}^{T} \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T \log\frac{2}{\delta}}.$$

The last step is to upperbound $\sum_{t=1}^{T} \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}$. For this purpose, Agrawal and Goyal [2013] rely on the analysis of Auer [2002] and the assumption that $\lambda \geq 1$. We provide an alternative approach using Cauchy-Schwartz inequality, Lemma 3, and Lemma 17 to get

$$\begin{aligned}
\sum_{t=1}^{T} \|\mathbf{x}_{a_t)}\|_{\boldsymbol{V}_t^{-1}} &\leq \sqrt{T \sum_{t=1}^{T} \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}} \\
&\leq \sqrt{T\left(2 + \frac{2}{\lambda}\right) \log\frac{|\boldsymbol{V}_T|}{|\boldsymbol{\Lambda}|}} \\
&\leq \sqrt{\frac{2 + 2\lambda}{\lambda} dT \log\left(1 + \frac{T}{K\lambda}\right)}.
\end{aligned}$$

Finally, we know that $E^{\hat{\boldsymbol{\alpha}}}(t)$ holds for all t with probability at least $1 - \frac{\delta}{2}$ and

$R'_T(t) = R_T(t)$ for all t with probability at least $1 - \frac{\delta}{2}$. Hence, with probability $1 - \delta$,

$$R_T \leq \frac{11g}{p}\sqrt{\frac{2+2\lambda}{\lambda}dT\log\left(1+\frac{T}{K\lambda}\right)} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}}+2\right)\sqrt{2T\log\frac{2}{\delta}}.$$

□

5.6 Regret bound of SpectralEliminator

The probability space induced by the rewards $r_1, r_2, \ldots$ can be decomposed as a product of independent probability spaces induces by rewards in each phase $[t_j, t_{j+1} - 1]$. Denote by $\mathcal{F}_j$ the σ-algebra generated by the rewards $r_1, \ldots, r_{t_{j+1}-1}$, i.e., received before and during the phase j. We have the following two lemmas for any phase j. Let $\overline{\boldsymbol{V}}_j = \boldsymbol{\Lambda} + \sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}_{a_s}\mathbf{x}_{a_s}^\intercal$ and $\hat{\boldsymbol{\alpha}}_j$ for $\hat{\boldsymbol{\alpha}}_{t_j}$.

Lemma 18. *For any fixed $\mathbf{x} \in \mathbb{R}^N$, any $\delta > 0$, and $\beta(\delta) = R\sqrt{2\log(2/\delta)} + \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$, we have for all j:*

$$\mathbb{P}\left(|\mathbf{x}^\intercal(\hat{\boldsymbol{\alpha}}_j - \boldsymbol{\alpha})| \leq \|\mathbf{x}\|_{\overline{\boldsymbol{V}}_j^{-1}}\beta(\delta)\right) \geq 1 - \delta$$

Proof. Defining $\boldsymbol{\xi}_j = \sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}_{a_s}\varepsilon_s$, we have:

$$|\mathbf{x}^\intercal(\hat{\boldsymbol{\alpha}}_j - \boldsymbol{\alpha})| = |\mathbf{x}^\intercal(-\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\Lambda}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\xi}_j)| \leq |\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\xi}_j| \tag{2.5}$$

The first term in the right hand side of (2.5) is bounded as:

$$\begin{aligned}
|\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\Lambda}\boldsymbol{\alpha}| &\leq \|\mathbf{x}^T\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\Lambda}^{1/2}\|\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{\alpha}\| \\
&= \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}\sqrt{\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\Lambda}\overline{\boldsymbol{V}}_j^{-1}\mathbf{x}} \\
&\leq \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}\sqrt{\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\mathbf{x}} = \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}\|\mathbf{x}\|_{\overline{\boldsymbol{V}}_j^{-1}}
\end{aligned}$$

Now consider the second term in the r.h.s. of (2.5). We have:

$$\left|\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\boldsymbol{\xi}_j\right| = \left|\sum_{s=t_{j-1}}^{t_j-1}(\mathbf{x}^\intercal\overline{\boldsymbol{V}}_j^{-1}\mathbf{x}_{a_s})\varepsilon_s\right|$$

Let us notice that the context vectors $(\mathbf{x}_{a_s})$ selected by the algorithm during phase $j-1$ only depend on their width $\|\mathbf{x}\|_{\boldsymbol{V}_s^{-1}}$ which does not depend on the rewards received during the phase $j-1$. Thus, given $\mathcal{F}_{j-2}$, the values $\mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}_{a_s}$ are deterministic for all $t_{j-1} \leq s < t_j$. Consequently, one may use a variant of Hoeffding bound for *scaled* sub-Gaussians [Wainwright, 2015], in particular for $\mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j = \sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}_{a_s} \varepsilon_s$, to get

$$\mathbb{P}\left(\left| \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j \right| \leq R \sqrt{2 \log\left(\frac{2}{\delta}\right) \sum_{s=t_{j-1}}^{t_j-1} \left(\mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}_{a_s} \right)^2} \right) \geq 1 - \delta$$

where ε_s is sub-Gaussian random variable and $\mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}_{a_s}$ is deterministic given $\mathcal{F}_{j-2}$. Further we deduce:

$$\mathbb{P}\left(\left| \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j \right| \leq R \sqrt{2 \log\left(\frac{2}{\delta}\right) \sum_{s=t_{j-1}}^{t_j-1} \left(\mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}_{a_s} \mathbf{x}_{a_s}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x} \right)} \right) \geq 1 - \delta$$

$$\mathbb{P}\left(\left| \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j \right| \leq R \sqrt{2 \log\left(\frac{2}{\delta}\right) \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \left(\sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}_{a_s} \mathbf{x}_{a_s}^\intercal \right) \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}} \right) \geq 1 - \delta$$

$$\mathbb{P}\left(\left| \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j \right| \leq R \sqrt{2 \log\left(\frac{2}{\delta}\right) \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \mathbf{x}} \right) \geq 1 - \delta$$

since $\overline{\boldsymbol{V}}_j^{-1}$ is symmetric and $\sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}_s \mathbf{x}_s^\intercal \prec \overline{\boldsymbol{V}}_j$. Thus:

$$\mathbb{P}\left(\left| \mathbf{x}^\intercal \overline{\boldsymbol{V}}_j^{-1} \boldsymbol{\xi}_j \right| \leq R \|\mathbf{x}\|_{\overline{\boldsymbol{V}}_j^{-1}} \sqrt{2 \log\left(\frac{2}{\delta}\right)} \right) \geq 1 - \delta$$

□

Lemma 19. *For all* $\mathbf{x} \in A_j$, $j > 1$, *we have:*

$$\min\left(1, \|\mathbf{x}\|_{\overline{\boldsymbol{V}}_j^{-1}}\right) \leq \frac{1}{t_j - t_{j-1}} \sum_{s=t_{j-1}}^{t_j-1} \min\left(1, \|\mathbf{x}_{a_s}\|_{\boldsymbol{V}_s^{-1}}\right)$$

Proof. Using Lemma 4, we have:

$$\begin{aligned}(t_j - t_{j-1}) \min\left(1,\, \|\mathbf{x}\|_{\overline{\boldsymbol{V}}_j^{-1}}\right) &\leq \max_{\mathbf{x}\in A_j} \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}\|_{\boldsymbol{V}_s^{-1}}\right) \\ &\leq \max_{\mathbf{x}\in A_{j-1}} \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}\|_{\boldsymbol{V}_s^{-1}}\right) \\ &\leq \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\, \max_{\mathbf{x}\in A_{j-1}} \|\mathbf{x}\|_{\boldsymbol{V}_s^{-1}}\right) \\ &= \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}_{a_s}\|_{\boldsymbol{V}_s^{-1}}\right),\end{aligned}$$

since the algorithm selects (during phase $j-1$) the arms with the largest width. □

Now we are ready to upper bound the cumulative regret of SpectralEliminator.

Proof of Theorem 5. Let $J = \lfloor \log_2 T \rfloor + 1$ and $t_j = 2^{j-1}$. We have:

$$\begin{aligned}R_T = \sum_{t=1}^{T} \mathbf{x}_{a^*}^\intercal \boldsymbol{\alpha} - \mathbf{x}_{a_t}^\intercal \boldsymbol{\alpha} &\leq 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min(2,\, \mathbf{x}_{a^*}^\intercal \boldsymbol{\alpha} - \mathbf{x}_{a_t}^\intercal \boldsymbol{\alpha}) \\ &\leq 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(2,\, \mathbf{x}_{a^*}^\intercal \hat{\boldsymbol{\alpha}}_j - \mathbf{x}_{a_t}^\intercal \hat{\boldsymbol{\alpha}}_j + \left(\|\mathbf{x}_*\|_{\overline{\boldsymbol{V}}_j^{-1}} + \|\mathbf{x}_t\|_{\overline{\boldsymbol{V}}_j^{-1}}\right) \beta(\delta')\right),\end{aligned}$$

in an event Ω of probability $1-\delta$, where we used Lemma 18 in the last inequality for $\delta' = \delta/(KJ)$. By definition of the action subset A_j at phase $j > 1$, under Ω, we have:

$$\mathbf{x}_{a^*}^\intercal \hat{\boldsymbol{\alpha}}_j - \mathbf{x}_{a_t} \hat{\boldsymbol{\alpha}}_j \leq \left(\|\mathbf{x}_{a^*}\|_{\overline{\boldsymbol{V}}_j^{-1}} + \|\mathbf{x}_{a_t}\|_{\overline{\boldsymbol{V}}_j^{-1}}\right) \beta(\delta'),$$

since $\mathbf{x}_{a^*} \in A_j$ for all $j \leq J$. By previous two lemmas and Cauchy-Schwarz inequality:

$$
\begin{aligned}
R_T &\leq 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(2,\, 4\beta(\delta')\|\mathbf{x}_{a_t}\|_{\overline{\boldsymbol{V}}_j^{-1}}\right) \\
&\leq 2 + (4\beta(\delta')+2) \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(1,\, \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}\right) \\
&\leq 2 + (4\beta(\delta')+2) \sum_{j=2}^{J} \frac{t_{j+1}-t_j}{t_j - t_{j-1}} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}\right) \\
&\leq 2 + (8\beta(\delta')+4) \sum_{j=2}^{J} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}_{a_t}\|_{\boldsymbol{V}_t^{-1}}\right) \\
&\leq 2 + (8\beta(\delta')+4) \sqrt{T \sum_{j=2}^{J} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\, \|\mathbf{x}_{a_t}\|^2_{\boldsymbol{V}_t^{-1}}\right)} \\
&\leq 2 + (8\beta(\delta')+4) \sqrt{T \sum_{j=2}^{J} 2 \log \frac{|\overline{\boldsymbol{V}}_j|}{|\boldsymbol{\Lambda}|}} \\
&\leq 2 + (8\beta(\delta')+4) \sqrt{2dT \log_2(T) \log\left(1 + \frac{T}{K\lambda}\right)}
\end{aligned}
$$

Finally, using $J = 1 + \lfloor \log_2 T \rfloor$, $\delta' = \delta/(KJ)$, and $\beta(\delta') \leq \beta(\delta/(K(1+\log_2 T)))$, we obtain the result of Theorem 5. □

Remark 7. *If we set* $\boldsymbol{\Lambda} = \mathbf{I}$ *in Algorithm 3 as in Remark 6, we get a new algorithm,* LinearEliminator*, which is a competitor to SupLinRel Auer [2002] and as a corollary to Theorem 5 also enjoys* $\tilde{\mathcal{O}}(\sqrt{DT})$ *upper bound on the cumulative regret. On the other hand, compared to SupLinRel,* LinearEliminator *and its analysis are significantly much simpler and elegant.*

6 Experiments

In this section, we compare empirical regret as well empirical computational complexity of SpectralTS, SpectralUCB, LinearTS, and LinUCB algorithms on artificial datasets with different types of underlying graph structures as well as on MovieLens and Flixster datasets. We do not include SpectralEliminator in our

experiments due to its impracticality for small time horizons since the algorithm updates confidence ellipsoid only at the end of the phase and therefore, the algorithm usually does not perform well for small time horizons. Moreover, we show an effect of reduced basis on both computational complexity and performance of the algorithms and the effect of Sherman-Morrison (computation of matrix inversions) and lazy updates (computation UCBs) on computational time. In all experiments we set both confidence parameter δ and noise variance R to 0.05. We did not include different values of δ and R since the results of the experiments are not sensitive to the values of these parameters.

6.1 Artificial datasets

To demonstrate the benefit of spectral algorithms we perform exhaustive experiments on artificial datasets with various underlying graphs structures. More precisely, we focus on problems where underlying graph structure forms lattice or is sampled either from the Barabási-Albert (BA) or Erdős-Rényi (ER) graph model. For all experiments on artificial datasets we set the number of arms N to 500 and time horizon T to 100. We created a random vector $\boldsymbol{\alpha}$ such that reward function $\mathbf{f} = \mathbf{Q}\boldsymbol{\alpha}$ is smooth on the graph. We did it by settings only the first 20 elements of $\boldsymbol{\alpha}$ to be nonzero. In order to be as objective as possible, we set the regularization parameter λ and confidence ellipsoid parameters v (TS) and c (UCB) respectively to the best empirical value. We ran algorithms with several different values of parameters and selected values which minimized average cumulative regret after several runs of algorithms. Figure 2.4 shows the dependence of cumulative regret on parameters with strong indications that SpectralTS and SpectralUCB can leverage smoothness of the reward function and outperform linear variants of algorithms.

6.1.1 Erdős-Rényi graph

For this experiment, we constructed an underlying graph as an Erdős-Rényi graph on 500 vertices with parameter 0.005 (probability of edge appearance). Values of the parameters used for the experiment are listed in Table 2.1. We selected the values for which the algorithms performed the best.

Figure 2.5a shows cumulative regrets of the algorithms with selected parameters. The regret of the spectral algorithms tends to be sublinear while regret of linear algorithms appears to be linear. Moreover, spectral algorithms reached much smaller

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.1$	$v = 0.1$	$\lambda = 1$	$c = 1$	$\lambda = 1$	$v = 0.1$	$\lambda = 0.1$	$c = 0.1$

Table 2.1: Best empirical parameters for BA graph model

regrets than their linear counterparts.

6.1.2 Lattice

For this experiment, we arranged 500 nodes to form a lattice and connected every pair of nodes by an edge if they are neighbors in the lattice. As in the case of other experiments, we selected empirically the best set of parameters (Table 2.2) and used them to plot cumulative regret of algorithms (Figure 2.5b). Even in this case, spectral algorithms performed well compared to the linear algorithms.

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.01$	$v = 0.1$	$\lambda = 0.1$	$c = 1$	$\lambda = 1$	$v = 0.1$	$\lambda = 0.1$	$c = 0.1$

Table 2.2: Best empirical parameters for BA graph model

6.1.3 Barabási-Albert graph

We constructed BA graph for the experiment in the following way. We started with k vertices ($k = 3$ in our case) without any connections between them. Then, we sequentially added one vertex at a time. Each new vertex was connected to $m \leq k$ previously added vertices and we sampled the connections according to the degrees of existing nodes; higher degree, bigger chance of the connection.

Table 2.3 summarizes the best empirical values of parameters for individual algorithms and Figure 2.5c shows the performance of algorithms for the parameters in Table 2.3. Here we can clearly see that spectral algorithms outperformed linear algorithm after just a few time steps. Note that empirically optimal parameters can sometimes be too aggressive and force an algorithm to exploit more than it should. This is probably the case of SpectralUCB algorithm in Figure 2.5c since the curve

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.001$	$v = 0.1$	$\lambda = 0.001$	$c = 0.01$	$\lambda = 0.01$	$v = 0.01$	$\lambda = 0.1$	$c = 0.1$

Table 2.3: Best empirical parameters for BA graph model

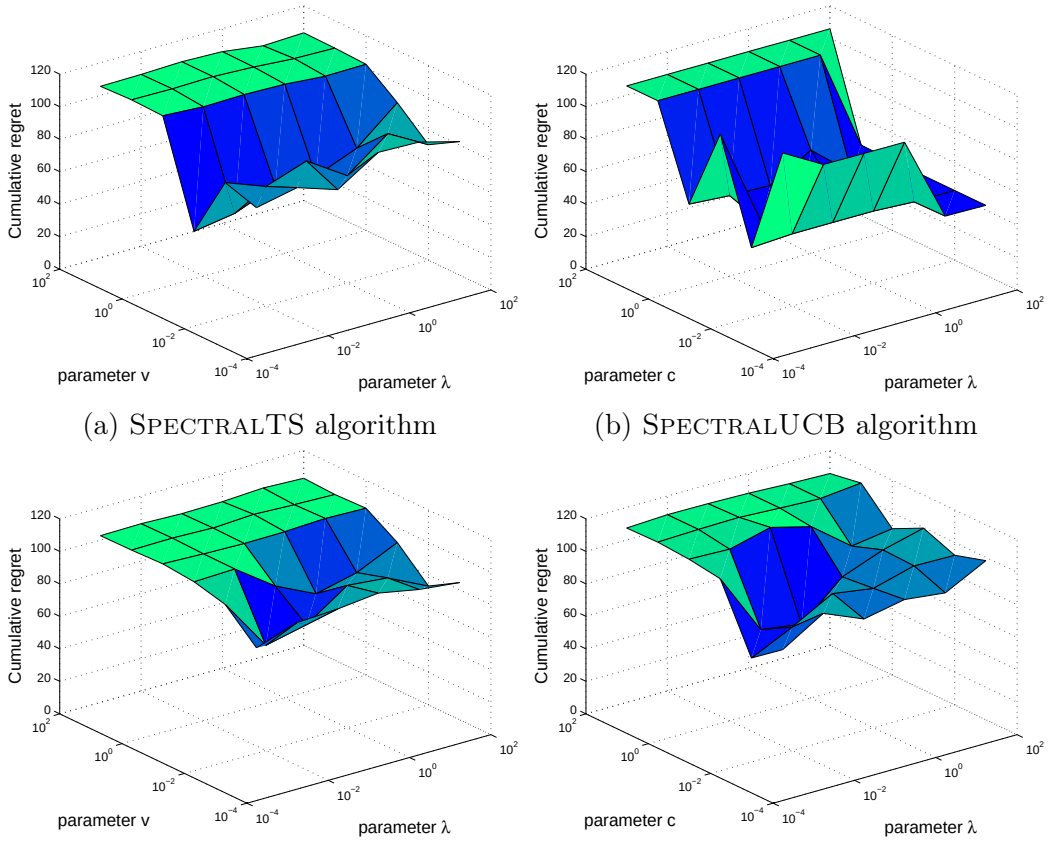
of cumulative regret of SpectralUCB appears to be linear for the time horizon used in our experiment. Therefore we included Figure 2.5d where we plotted cumulative regret of SpectralUCB for empirically suboptimal value of $c = 1$ (close to the theoretical value of c) to demonstrate sublinear tendencies of the regret.

6.2 Effect of smoothness on regret

In this section, we show an effect of the smoothness of the reward function on the performance of spectral algorithms. We used BA graph on 500 vertices for the experiment with time horizon 100. The value of the effective dimension is roughly 8. We controlled the smoothness by explicitly setting the number of eigenvectors used for constructing reward function; by letting 5, 25, 100 or 500 elements of $\boldsymbol{\alpha}$ to be nonzero. Note that value of the effective dimension is the same for every reward function we used, since the definition of effective dimension is independent of the reward function. Table 2.4 shows how smoothness changes with number of nonzero elements of $\boldsymbol{\alpha}$ and Figures 2.6a and 2.6b show that spectral algorithm are able to leverage spectral properties of underlying graph better if the reward function is smoother. This is also supported by the theory since in our experiment, smoothness of the reward function decreases with the smaller number of eigenvectors and therefore, regret bounds of the spectral algorithms are decreasing as well.

Number of nonzero components	5	25	100	500
Smoothness of reward function ($\boldsymbol{\alpha}^\intercal \boldsymbol{\Lambda} \boldsymbol{\alpha}$)	1.56	11.16	58.12	216.89
Regret of SpectralTS	7.99	32.80	94.10	123.79
Regret of SpectralUCB	3.05	22.84	108.19	130.54

Table 2.4: Effect of smoothness on regret



(a) SpectralTS algorithm (b) SpectralUCB algorithm

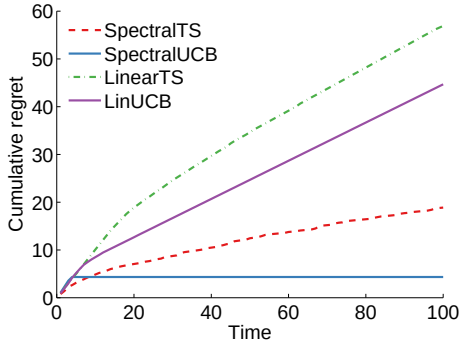
(c) LinearTS algorithm (d) LinUCB algorithm

Figure 2.4: Dependence of cumulative regret on confidence and regularization parameters v and c

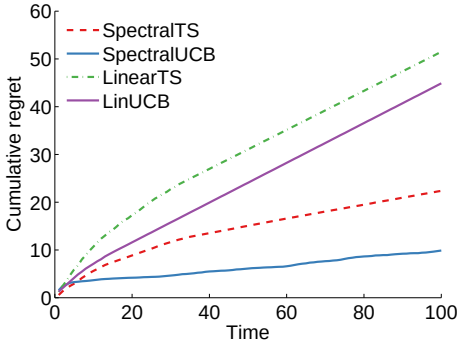
6.3 Computational complexity improvements

In general computation of N UCBs is computationally more expensive than sampling in TS. In Section 4.4 we discussed several possibilities to speed up algorithms. The impact of lazy updates for computing UCBs and effect of Sherman-Morrison formula on matrix inversion is demonstrated in Figure 2.7. The plot clearly shows that lazy updates can improve computational time of UCB to the point where the computational time of SpectralUCB is comparable, in some cases even better than the computational time of SpectralTS.

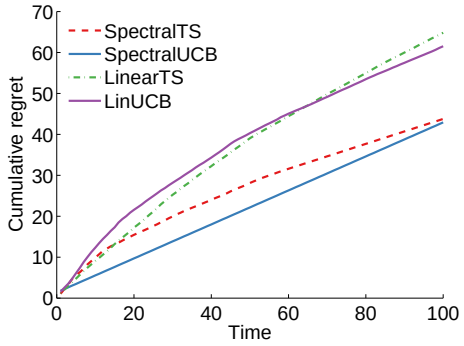
Another possible computational time improvement, discussed in Section 4.4, is by extracting only first $L \ll N$ eigenvectors of the graph Laplacian. First, the computational complexity of such operation is $\mathcal{O}(Lm \log m)$, where m is the number of



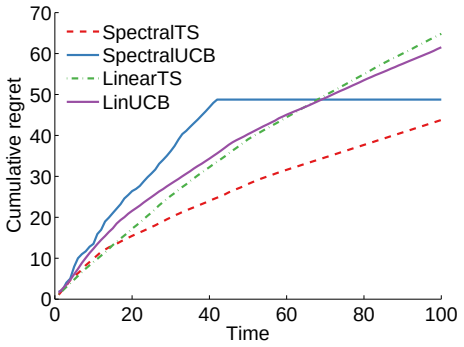
(a) Erdős-Rényi graph



(b) Lattice

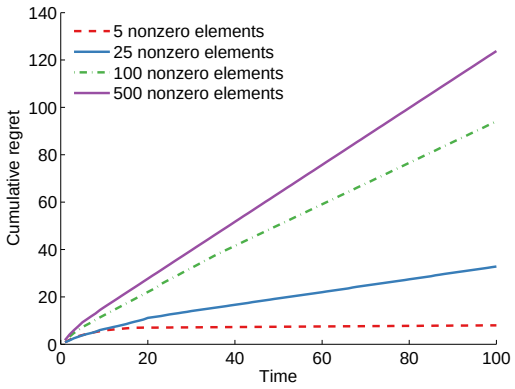


(c) Barabási-Albert graph

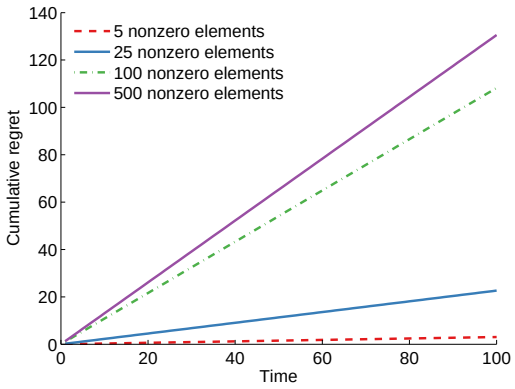


(d) Barabási-Albert graph with suboptimal parameters for SpectralUCB

Figure 2.5: Cumulative regret comparison of algorithms for different underlying graphs



(a) SpectralTS algorithm



(b) SpectralUCB algorithm

Figure 2.6: Cumulative regret of SpectralTS and SpectralUCB for reward functions with different smoothness

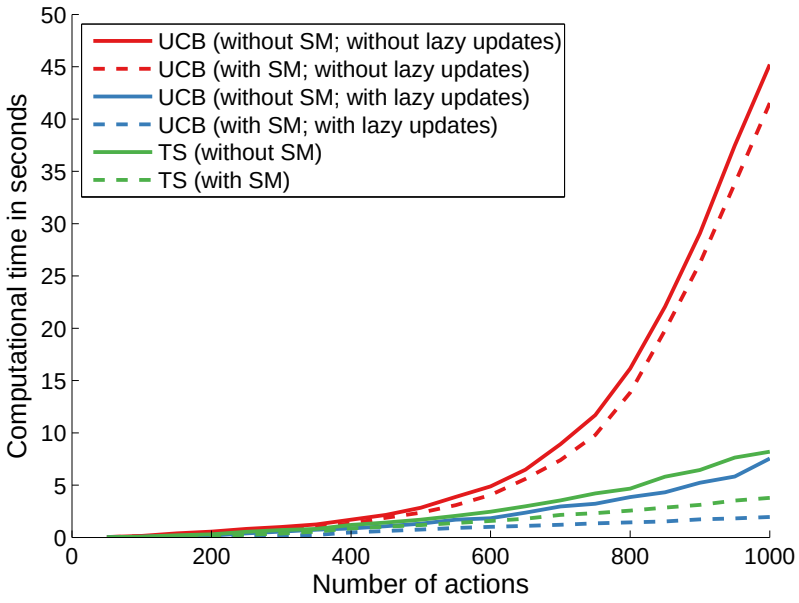


Figure 2.7: Impact of lazy updates and Sherman-Morrison formula on computational time

edges. Second, the least-squares problem that we have to do in each time step of the algorithm is only L dimensional. In Figure 2.8 we plot the cumulative regret and the total computational time in seconds (log scale) of SpectralUCB algorithm for a single user from the MovieLens dataset. We varied L as 20, 200, and 2000 which corresponds to about 1%, 10% and 100% of basis functions ($N = 2019$). The total computational time also includes the computational savings from lazy updates and iterative matrix inversion. We see that with 10% of the eigenvectors we can achieve similar performance as for the full set for the fraction of the computational time.

6.4 MovieLens experiments

In this experiment, we took user preferences and the similarity graph over movies from the MovieLens dataset [Lam and Herlocker, 2012], a dataset of 6k users who rated one million movies. Firstly, we extracted a subset of 400 users and 618 movies with at least 500 ratings. Then we divided the dataset into three parts. The first is used to build our model of users, the rating that user i assigns to movie j. We factor the user-item matrix using low-rank matrix factorization [Keshavan et al., 2009] as $\mathbf{M} \approx \mathbf{U}\boldsymbol{V}'$, a standard approach to collaborative filtering. The rating that user i assigns to movie j is estimated as $\hat{r}_{i,j} = \langle \mathbf{u}_i, \mathbf{v}_j \rangle$, where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$ and $\mathbf{v}_j$

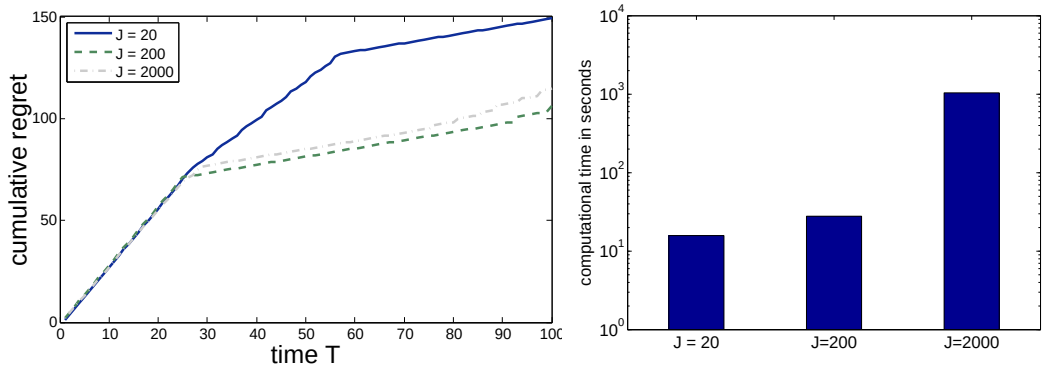


Figure 2.8: Regret and computational time of SpectralUCB with reduced basis

is the j-th row of $\boldsymbol{V}$. The rating $\hat{r}_{i,j}$ is the payoff of pulling arm j when recommending to user i.

The second part of the dataset is used for parameter estimation. Similarly to the case of the first part, we completed ratings using low-rank factorization. We used a different part of the dataset in order to avoid dependencies.

The last part of the dataset is used to build our similarity graph over movies. We factor the dataset in the same way as the first two parts of the dataset. The graph contains an edge between movies i and i' if the movie i' is among 5 nearest neighbors of the movie i in the latent space of items $\boldsymbol{V}$. The weight on all edges is one. Notice that if two items are close in the item space, then their expected rating is expected to be similar. However, the opposite is not true. If two items have a similar expected rating, they do not have to be close in the item space. In other words, we take advantage of ratings but do not hardwire the two similarly rated items to be similar.

Table 2.5 summarizes the best parameters learned on training part of the dataset. We used the parameters to run algorithms on test part of the dataset. Figure 2.9a shows 20 random users sampled from the testing part of the MovieLens dataset. We evaluated the regret of all four algorithms for $T = 500$ and compared the computational time of the algorithms. The results show us several interesting observations. First, spectral algorithms are consistently outperforming linear algorithms. Second, as we mentioned in Section 4.4, we use lazy updates for UCB algorithms which can improve computational time significantly. We can see that in our experiment, computational time of UCB algorithms was better than computational time of TS

algorithms even though in general, TS algorithms are computationally more efficient than UCB algorithms without lazy updates.

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.001$	$v = 0.1$	$\lambda = 0.1$	$c = 1$	$\lambda = 100$	$v = 1$	$\lambda = 0.001$	$c = 0.001$

Table 2.5: Best empirical parameters for Movielens dataset

6.5 Flixster experiments

We also performed experiments on users preferences from the movie recommendation website Flixster. The social network of the users was crawled by Jamali and Ester [2010] and then clustered by Graclus [2013] to obtain a strongly connected subgraph. Similarly like in the case of Movielens, we extracted a subset of users and movies, where each movie has at least 500 ratings. This resulted in a dataset of 972 movies and 1070 users. As with MovieLens dataset, we completed the missing ratings by a low-rank matrix factorization and used it construct a 5-NN similarity graph.

Again in Figure 2.9b, we sampled 20 random users and evaluated the regret of all four algorithms for $T = 50$.

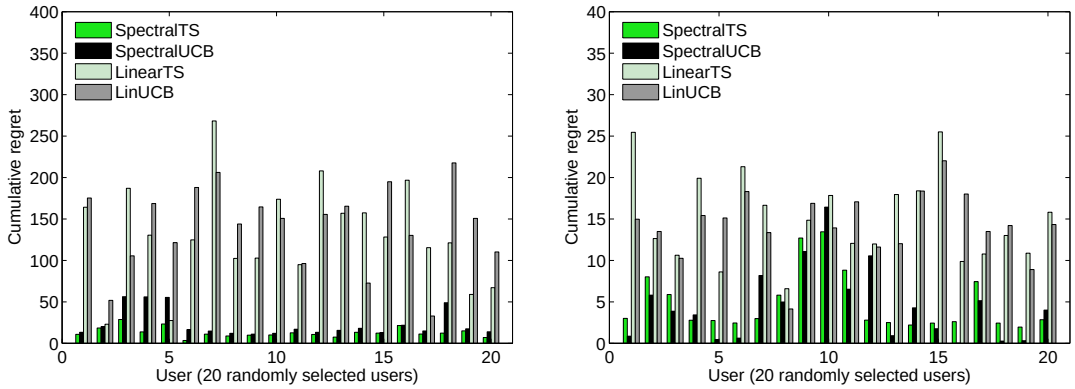
Similarly like in the case of MovieLens dataset, we set parameter λ to 0.01 while setting the parameter v of SpectralTS to be ten times smaller than the theoretical value.

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.01$	$v = 0.1$	$\lambda = 0.01$	$c = 0.11$	$\lambda = 1$	$v = 0.1$	$\lambda = 1$	$c = 1$

Table 2.6: Best empirical parameters for Flixster dataset

6.6 Experiment design modifications

While performing experiments we found several ways to adjust the experiment design in order to improve the performance of the algorithms.



(a) Movielens dataset, cumulative regret for 20 randomly selected users

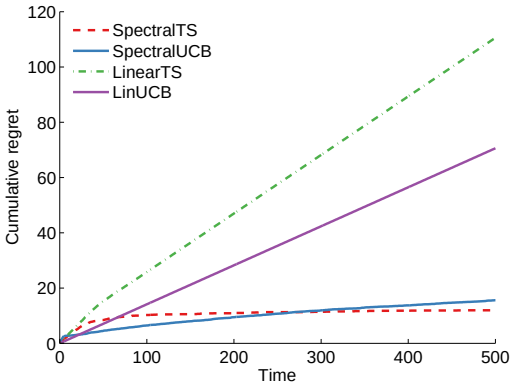
(b) Flixster dataset, cumulative regret for 20 randomly selected users

Figure 2.9: Comparison of spectral and linear algorithms

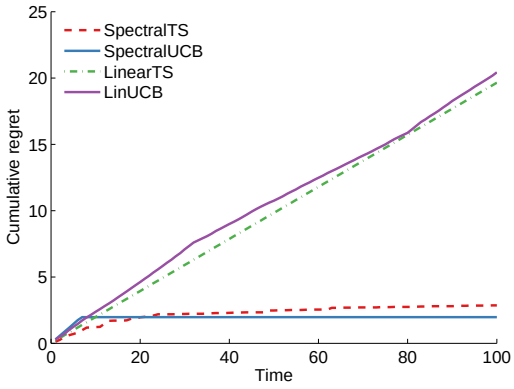
- **Adjusting the number of edges in the graph.** Usually, real world datasets do not come with a graph structure. Therefore, we usually construct the nearest neighbor graph which connects only the most similar actions. By reducing the number of neighbors we are increasing the effective dimension (worsening of the regret bound) and decreasing smoothness of the function (improving the regret bound). Finding good trade off and adjusting the number of the edges can improve the performance of the algorithms significantly.

- **Scaling confidence ellipsoid** (parameter c in SpectralUCB and parameter v in SpectralTS)**.** Usually, the algorithms are too conservative and the bounds are too loose in order to prove high probability bounds. Therefore, reducing the size of the confidence ellipsoid can sometimes improve the performance of the algorithm at the price that some bounds might not hold anymore. In fact, in our experiments, we did not use theoretical values of confidence parameters. Instead, we used the values for which the algorithms had good empirical performance.

- **Magnitude of regularization parameter λ.** By setting λ to a large value, all regularized eigenvalues become similar and therefore the algorithms take the graphs structure less into account. On the other hand, if the regularization parameter λ is small, the algorithms follow the graph structure more. Therefore, it in order to leverage the graphs structure the algorithms have to find a good compromise while setting λ. In our experiments, we tried several values of λ and picked the value with the best empirical performance.



(a) Movielens dataset, cumulative regret for one random user



(b) Flixster dataset, cumulative regret for one random user

Figure 2.10: Comparison of spectral and linear algorithms

- **Scaling of the graphs.** By scaling all the weights of the graph by some constant we scale the gap between the eigenvalues and therefore changing the value of the effective dimension. Moreover, by scaling weights we are also changing the smoothness of the reward function. Therefore, scaling the weights change the emphasis of the algorithm on the graph.

Chapter 3

Bandits with side observations

In this chapter, we take a closer look at the multi-armed bandit problem with side observations (Section 3.2 in Chapter 1). The structure of the problem is represented as a graph on top of the action set. In this framework, we assume that on top of the bandit feedback, the learner observes losses of the neighbors of the selected action. Depending on an application, these side observations may be of different quality. Inspired by several real-world applications, we introduce several extensions to bandits and look at some of the solutions solving the problems.

This chapter is structured in the following way. First, we introduce the side-observation framework. Second, we present our approach to the exploration-exploitation dilemma, called implicit exploration, and we demonstrate the idea on the basic Exp3 algorithm for the basic adversarial bandits. The main part of the chapter explores several settings, following side-observations framework, driven by real world applications. For each setting, we provide formal definition and algorithm with theoretical, in some cases also empirical, guarantees. In the last part of the chapter, we present the proofs of the main results for each setting.

Contents

1 Framework of bandits with side observations

A general framework of multi-armed bandit problem with side observations is motivated by applications involving graphs [Mannor and Shamir, 2011, Alon et al., 2013, Kocák et al., 2014a, Alon et al., 2015]. In this framework, the learner faces an online

Input:
  Known set of actions $[N]$
  Time horizon T (not necessarily known)
for $t = 1$ **to** T **do**
  The environment (adversary) chooses a loss function over the arms, with $\ell_{t,i}$ being the loss associated with arm $i \in [N]$ at time t
  The environment chooses an underlying observability graph G_t. **Assumptions on G_t may vary for different settings**
  The learner chooses an action $I_t \in [N]$.
  The learner suffers loss ℓ_{t,I_t} of the action I_t
  The learner observes losses of neighbors of I_t according to G_t
  The learner observes some part of G_t. **"Part" depends on setting**
end for
Goal: Minimize cumulative regret $R_t = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right]$

Figure 3.1: General framework of bandits with side observations

learning problem consisting of N actions. In every round, the learner chooses an action and incurs (also observes) the loss corresponding to the chosen action. Moreover, the losses of some additional actions are revealed to the learner as well. These side observations are specified by a directed observability graph with actions as nodes; playing action i reveals loss of action j if there is an edge from i to j. This framework is described in Figure 3.1.

Later in this chapter, we study a variety of different settings following this framework. These settings capture different problems with various assumptions on the graph structure and the quality of the side observations. Namely, we study following four settings.

In Section 2, we consider a well studied setting of Mannor and Shamir [2011]. This setting contains as few assumptions as possible. The graph generated by the environment can be arbitrary with no assumptions, possibly even chosen by an adversary. This particular setting is not new, therefore, there exist several algorithms for the problem. However, the mot of the algorithms have one issue. In order to show strong, non-trivial guarantees, the algorithms need access to the graph before the action is taken. This might cause problems for some applications. We solve this problem with a novel approach to the exploration, called implicit exploration and present an algorithm with optimal theoretical guarantees while having access to the graph only the actions is taken. Recent result by Cohen et al. [2016] shows that without an access

to the observability graph, at least after playing the action, there is no hope to prove any nontrivial guarantees for the algorithms. More precisely, the learner needs to observe at least second neighborhood of the selected action. Without access to the graph, the only solution is to neglect all the side observations and use an algorithm for bandits with its guarantees.

In Section 3, we consider another simple setting with only a few assumptions. We are inspired by small social groups where all the members are equal and the additional information is obtained from every member with the same probability. We model this problem using a graph sampled from Erdős-Rényi distribution with parameter r_t; every edge in the graph appears with probability r_t independently of each other. In fact, this assumption enables us to design an algorithm with strong theoretical guarantees without accessing the graph, even if r_t is controlled by an adversary. Moreover, thanks to the constraint on the graph, the algorithm presented in the section is the first algorithm using side observations, in a non-trivial way, without access to the graph.

In Section 4, we consider a setting with noisy side observations. In some problems, side observations are not perfect (e.g. sensor networks). We model this problem using a weighted graph where the weights represent the amount of information obtained from the neighbors. In other words, a weight can be seen as an information to noise ratio of a side observation. Similarly like in Section 2, we need the assumption that the learner observes substantial part of the graph, at least after the action is taken, in order to obtain non-trivial theoretical guarantees.

In Section 5, we focus on the problem where an actions is more complex (e.g. packet routing, action consist of a path) while obtaining some side observations (e.g. information about a path with a shared sub-path). We model this problem as a combinatorial semi-bandit problem with side observations. This setting directly extends combinatorial bandits to the side observation scenario where the learner plays a combinatorial structure (e.g. path, clique, circle, component) consisting of possibly many nodes of the graph, receiving loss of all nodes contained in the structure played, and observing losses of all the neighbors of the nodes contained in the played structure. In this setting, graphs can be completely adversarial but we still need to assume that at least second neighborhood of played nodes is revealed to the learner in order to get nontrivial theoretical results.

1.1 Existing algorithms and results

Since the amount of feedback in bandits with side observations interpolates between bandit feedback and full-information feedback, a regret bound of an algorithm solving the problem should interpolate between $\widetilde{\mathcal{O}}(\sqrt{T})$ (full-information) and $\widetilde{\mathcal{O}}(\sqrt{NT})$ (bandits), depending on the observability graph. Mannor and Shamir [2011] showed $\Omega(\sqrt{\alpha T})$ lower bound for fixed undirected graphs, where α is an independence number of the graph. Later, Alon et al. [2013] extended this lower bound to the case of fixed directed graphs, where α is the independence number of the graph (size of the largest independent set of nodes; nodes that are not connected by any edge). This lower bound is very natural for the problem since restricting the action set of the problem to an independence set of size α, the problem become equivalent to the bandit problem on this set with a lower bound of $\Omega(\sqrt{\alpha T})$.

The first algorithm for the setting was presented in [Mannor and Shamir, 2011]. This algorithm is called ELP, achieves regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$ in the case of undirected graph, and regret bound of $\widetilde{\mathcal{O}}(\sqrt{\chi(G)T})$ in the case of directed graph, where $\chi(G)$ is a clique number of graph G (the smallest number partitions of G such that every partition is a clique). However, in order to achieve the bound, ELP algorithm needs to compute linear program to precisely define exploration distribution for the algorithm. This comes with two drawbacks. First, the linear program can be computationally expensive. Second, in order to compute linear program, the learner needs to have an access to the graph before his decision. Later, Alon et al. [2013] introduced two new algorithms for the setting. The first algorithm is called Exp3-SET and is designed for undirected graphs. This algorithm achieves optimal regret bound of order $\Omega(\sqrt{\alpha T})$ but, unlike ELP algorithm, it does not need to know observability graph in advance and does not need to compute any linear program. The only requirement is that the graph is revealed to the algorithm after playing an action. The second algorithm is called Exp3-DOM and is designed for directed graphs. This algorithm achieves regret bound of order $\Omega(\sqrt{\alpha T})$ which matches lower bound and improves the bound of ELP algorithm. However, in order to show the regret bound, the algorithm needs to find the smallest dominating set (set of nodes such that: there is an edge from a node in the set to every node outside of the set) which is an NP-hard problem. Therefore, in practice, the algorithm uses only a greedy approximation of a dominating set which makes the algorithm computationally efficient for the price of extra log factor in the regret bound coming from the approximation.

In order to achieve optimal regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$, previously mentioned algorithms need to encourage exploration and guarantee that every arm is played some-

times. This is mostly done by mixing the probability distribution of the learner with some carefully selected exploration distribution. To define this exploration distribution, the algorithms usually need to compute a linear program or to find a dominating set which could be very expensive. The first computationally efficient algorithm for the directed case was EXP3-IX algorithm (Section 2), introduced in [Kocák et al., 2014a]. This algorithm uses a novel approach to exploration which we call implicit exploration. The regret bound for this algorithm is of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$ and the algorithm does not need an access to the graph before playing an action thanks to a new approach to exploration called *implicit exploration*. The details concerning this algorithm are presented later in Chapter 3. Shortly after introducing EXP3-IX, Alon et al. [2015] came with EXP3.G algorithm also achieving the same regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha T})$. This algorithm is using a carefully tuned mixing with uniform distribution on all the actions, this makes the algorithm computationally efficient as well. Furthermore, Alon et al. [2015] considered a strictly more difficult setting than ours, where the loss of the chosen action may not be a part of the received feedback.

1.2 Exploration in EXP3-based algorithms

Before diving into the bandits with side observations, we start by several variations of basic EXP3 algorithm. We use them to demonstrate basic approaches of EXP3-based algorithms to exploration and loss/reward estimation.

EXP3 is the most popular approach to solve the adversarial multi-armed bandit problem. This algorithm was introduced in Auer et al. [2002b] and is based on two simple ideas. Since the learner plays an action I_t, at time, according to a probability distribution $\boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,N})$, it is possible to construct unbiased loss estimate $\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{p_{t,i}} \mathbb{1}\{I_t = i\}$ for every actions $i \in [N]$, even if the action is not played. The second idea is to use exponential weights [Littlestone and Warmuth, 1994], constructed from cumulative loss estimates, to define the new probability distribution over the arms.

The first analysis of EXP3 algorithm was provided by Auer et al. [2002b]. The algorithm was designed with rewards instead of losses and, as it turned out, the algorithm did not explore enough in order to show strong theoretical guarantees. The intuition behind this problem is simple: playing an arm results in high reward estimate of the arm while all the other estimates are zero. This means that the probability of playing this arm will increase in the next round even more. This results in decreasing the amount of exploration. In other words, the arms which were

played often in the past are more likely to be played in the future. The simplest and most used way to fix this issue is to dedicate some of the rounds to exploration; sampling an action from a uniform distribution. In practice, the algorithm encourages the exploration by mixing the probability distribution with a uniform distribution on the set of arms. More precisely, the learner plays according to the probability distribution $\boldsymbol{p}'_t = (p'_{t,1}, \ldots, p'_{t,N})$ such that

$$p'_{t,i} = (1 - \gamma_t)p_{t,i} + \frac{\gamma_t}{N},$$

where $\gamma_t \in [0, 1]$ is a parameter controlling the amount of exploration.

Later it turned out that mixing is not necessary if the algorithm uses losses instead of rewards. The reason is again simple; if an arm is played, the loss estimate of this arm can be big compared to the zero loss estimates of the other arms. Therefore, by playing an action, we decrease the probability of playing this action again in the next round, and therefore encouraging the exploration indirectly. In fact, using losses instead of rewards one can prove strong theoretical guarantees on expected regret, even without explicit mixing. However, even using losses, there was a common belief that extra exploration is necessary to obtain strong high-probability bounds on the regret [Auer et al., 2002b, Audibert and Bubeck, 2010, Beygelzimer et al., 2011, Bubeck and Cesa-Bianchi, 2012].

Another instance of the problem where extra exploration is necessary is a multi-armed bandit problem with side observations [Mannor and Shamir, 2011, Alon et al., 2013]. Moreover, it is necessary to have an extra exploration in this problem; even for regret bounds in expectation. We look at this problem more closely later since it forms a basis for the problems presented this chapter.

As we see, controlling exploration is a substantial part of the algorithms for adversarial bandit problems. Even though the mixing proved to be an efficient way to deal with this problem, it might be impractical to use mixing in some problems. Combinatorial bandits represent a good example since, in general, designing a mixing probability distribution can be computationally very expensive. Therefore, we come with a different approach to the exploration which we call **I**mplicit e**X**ploration, in short **IX** [Kocák et al., 2014a]. Instead of explicitly mixing the probability distribution of the learner, implicit exploration introduces a bias to the loss estimates which controls exploration indirectly.

1.3 Implicit exploration and EXP3 algorithm

In this section, we present the idea of implicit exploration. Even though the basic EXP3 algorithm with losses does not need any additional exploration, for the simplicity, we use this algorithm to demonstrate the idea of implicit exploration. We show later applications of this idea which enable us to construct efficient algorithms for more complex problems.

In explicit exploration, the learner is mixing his probability distribution with an exploration distribution. On the other hand, the main idea of implicit exploration is biasing loss estimates to encourage exploration. Usual loss estimates used in EXP3 algorithm are unbiased and in the following form

$$\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{p_{t,i}} \mathbb{1}\{I_t = i\},$$

where I_t is an action chosen by the learner and $p_{t,i}$ is the probability of playing an action i at time t; $\mathbb{P}[I_t = i] = p_{t,i}$. The idea of implicit exploration is to introduce a small bias term $\gamma_t \geq 0$ in the loss estimates:

$$\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{p_{t,i} + \gamma_t} \mathbb{1}\{I_t = i\}.$$

We call γ_t the implicit exploration term. The essential steps of basic EXP3 algorithm with implicit exploration are following:

1. Construct exponential weights using loss estimates

$$w_{t,i} = \frac{1}{N} \exp\left(-\eta_t \textstyle\sum_{s=1}^{t-1} \hat{\ell}_{s,i}\right) \qquad \text{for all} \quad i \in [N]$$

2. Create a probability distribution $p_t = (p_{t,1}, \ldots, p_{t,N})$ such that

$$p_{t,i} = \frac{w_{t,i}}{W_t} \qquad \text{where} \qquad W_t = \textstyle\sum_{i=1}^{N} w_{t,i}$$

3. Play an action $I_t \sim p_t = (p_{t,1}, \ldots, p_{t,N})$ and incur the loss ℓ_{t,I_t} of the action

4. Construct loss estimates

$$\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{p_{t,i} + \gamma_t} \mathbb{1}\{i = I_t\} = \begin{cases} \frac{\ell_{t,i}}{p_{t,i}+\gamma_t}, & \text{if } i = I_t \\ 0, & \text{otherwise} \end{cases}$$

The only difference compared to the basic Exp3 algorithm is Step 4, where the loss estimates are constructed with an extra γ_t term. Since this algorithm forms a basis for most of the work in this chapter, we show the most important steps of its analysis in the next section.

1.4 Exp3-based algorithms

Most of the algorithms in this chapter are based on Exp3 algorithm by Auer et al. [2002b] for adversarial bandits. These algorithms follow an algorithms template described in Algorithm 4.

Algorithm 4 can be used in various sequential learning problems where the learner plays a single action. Usually, the performance of the algorithm depends on learning rates η_t and loss estimates $\hat{\ell}_{t,i}$. In general, these quantities are problem and algorithm dependent. Later in this chapter, we show several different settings and algorithms following this template.

1.4.1 Analysis of Exp3-based algorithms

To acquire a deeper understanding of the problem, we present an analysis of algorithms based on Exp3 and point out some standard results and challenges that come from different settings. The first part of the analysis is the same for the most of the Exp3-based algorithm and therefore, it will be useful to formulate it as the following lemma.

Lemma 20. *For all $t \in [T]$, let η_t is a positive learning rate such that $\eta_{t+1} \leq \eta_t$ and $\ell_{t,i}$ is a non-negative loss of action i at time t. For algorithm following* Exp3 *template describes in Algorithm 4 we have*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right]$$

Algorithm 4 Algorithm template: EXP3 [Auer et al., 2002a]

Input:
  Set of actions [N]
  Not necessarily known time horizon T
Initialization:
  Set initial cumulative loss estimates $\widehat{L}_{0,i} = 0$ for all $i \in [N]$
for $t = 1$ **to** T **do**
  Select learning rate η_t
  Construct exponential weight

$$w_{t,i} = \frac{1}{N} \exp\left(-\eta_t \widehat{L}_{t-1,i}\right) \qquad \text{for all} \quad i \in [N]$$

  Create a probability distribution $p_t = (p_{t,1}, \ldots, p_{t,N})$ such that

$$p_{t,i} = \frac{w_{t,i}}{W_t} \qquad \text{where} \qquad W_t = \textstyle\sum_{i=1}^{N} w_{t,i}$$

  Choose an action $I_t \sim p_t = (p_{t,1}, \ldots, p_{t,N})$ to play
  Observe loss of I_t and possibly some additional observations
  Using observations construct loss estimates $\hat{\ell}_{t,i}$ for all $i \in [N]$
  Update cumulative loss estimates $\widehat{L}_{t,i} = \widehat{L}_{t-1,i} + \hat{\ell}_{t,i}$ for all $i \in [N]$
end for

holds for any $j \in [N]$ where the expectation is taken with respect to the randomness of the learner as well as the randomness of the environment.

Note that if the bias of our loss estimates is close to zero, the left-hand side of the bound is closely related to the regret. This is due to the fact that the first term is close to the expected loss of the learner while the second term is close to the cumulative loss of the best arm.

Proof. The proof of this lemma follows a standard analysis of EXP3 algorithm with adaptive learning rate, e.g. Lemma 1 of Györfi and Ottucsák [2007]. We start by introducing some notation. Let

$$\widehat{L}_{t-1,i} = \sum_{s=1}^{t-1} \hat{\ell}_{s,i}, \qquad W_t = \frac{1}{N} \sum_{i=1}^{N} e^{-\eta_t \widehat{L}_{t-1,i}}, \qquad W'_t = \frac{1}{N} \sum_{i=1}^{N} e^{-\eta_{t-1} \widehat{L}_{t-1,i}}.$$

Next, we track the evolution of $\log W'_{t+1}/W_t$ to control the regret. We have

$$\begin{aligned}\frac{1}{\eta_t}\log\frac{W'_{t+1}}{W_t} &= \frac{1}{\eta_t}\log\sum_{i=1}^{N}\frac{\frac{1}{N}e^{-\eta_t\widehat{L}_{t,i}}}{W_t} = \frac{1}{\eta_t}\log\sum_{i=1}^{N}\frac{w_{t,i}e^{-\eta_t\hat{\ell}_{t,i}}}{W_t}\\ &= \frac{1}{\eta_t}\log\sum_{i=1}^{N}p_{t,i}e^{-\eta_t\hat{\ell}_{t,i}} \le \frac{1}{\eta_t}\log\sum_{i=1}^{N}p_{t,i}\left(1-\eta_t\hat{\ell}_{t,i}+\frac{1}{2}(\eta_t\hat{\ell}_{t,i})^2\right)\\ &= \frac{1}{\eta_t}\log\left(1-\eta_t\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i}+\frac{\eta_t^2}{2}\sum_{i=1}^{N}p_{t,i}(\hat{\ell}_{t,i})^2\right),\end{aligned}$$

where we used the inequality $\exp(-x) \le 1 - x + x^2/2$ that holds for $x \ge 0$.

Using the inequality $\log(1-x) \le -x$ that holds for all x, we get

$$\begin{aligned}\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i} &\le \left[\frac{\log W_t}{\eta_t}-\frac{\log W'_{t+1}}{\eta_t}\right]+\sum_{i=1}^{N}\frac{\eta_t}{2}p_{t,i}(\hat{\ell}_{t,i})^2\\ &= \left[\left(\frac{\log W_t}{\eta_t}-\frac{\log W_{t+1}}{\eta_{t+1}}\right)+\left(\frac{\log W_{t+1}}{\eta_{t+1}}-\frac{\log W'_{t+1}}{\eta_t}\right)\right]+\sum_{i=1}^{N}\frac{\eta_t}{2}p_{t,i}(\hat{\ell}_{t,i})^2.\end{aligned}$$

The second term in brackets on the right-hand side can be bounded as

$$W_{t+1}=\sum_{i=1}^{N}\frac{1}{N}e^{-\eta_{t+1}\widehat{L}_{t,i}}=\sum_{i=1}^{N}\frac{1}{N}\left(e^{-\eta_t\widehat{L}_{t,i}}\right)^{\frac{\eta_{t+1}}{\eta_t}}\le\left(\sum_{i=1}^{N}\frac{1}{N}e^{-\eta_t\widehat{L}_{t,i}}\right)^{\frac{\eta_{t+1}}{\eta_t}}=(W'_{t+1})^{\frac{\eta_{t+1}}{\eta_t}},$$

where we applied Jensen's inequality to the concave function x^{η_{t+1}/η_t} for $x \in \mathbb{R}$. The function is concave since $\eta_{t+1} \le \eta_t$ by definition. Taking logarithms in the above inequality, we get

$$\frac{\log W_{t+1}}{\eta_{t+1}}-\frac{\log W'_{t+1}}{\eta_t}\le 0.$$

Using this inequality, we prove a standard inequality which arises in the most of the proofs of algorithms based on Exp3 algorithm.

$$\sum_{i=1}^{N}p_{t,i}\hat{\ell}_{t,i}\le\frac{\eta_t}{2}\sum_{i=1}^{N}p_{t,i}\left(\hat{\ell}_{t,i}\right)^2+\left(\frac{\log W_t}{\eta_t}-\frac{\log W_{t+1}}{\eta_{t+1}}\right)$$

Summing up both sides over the time and taking expectations, we get

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}\left(\hat{\ell}_{t,i}\right)^2\right] + \mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{\log W_t}{\eta_t} - \frac{\log W_{t+1}}{\eta_{t+1}}\right)\right].$$

The second term on the right-hand side telescopes into

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{\log W_t}{\eta_t} - \frac{\log W_{t+1}}{\eta_{t+1}}\right)\right] &= \mathbb{E}\left[\frac{\log W_1}{\eta_1} - \frac{\log W_{T+1}}{\eta_{T+1}}\right] \\
&= \mathbb{E}\left[-\frac{\log \sum_{i=1}^{N} w_{T+1,i}}{\eta_{T+1}}\right] \\
&\leq \mathbb{E}\left[-\frac{\log w_{T+1,j}}{\eta_{T+1}}\right] \\
&= \mathbb{E}\left[\frac{-1}{\eta_{T+1}}\log\left(\frac{1}{N}e^{-\eta_{T+1}\widehat{L}_{T,j}}\right)\right] \\
&= \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\widehat{L}_{T,j}\right].
\end{aligned}$$

Applying this bound to the previous inequality concludes the proof □

The bound in Lemma 20 holds for every algorithm based on Exp3 and gives us

$$\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right]}_{A} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\hat{\ell}_{t,j}\right]}_{B} \leq \underbrace{\mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right]}_{C} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right]}_{D}. \tag{3.1}$$

The next part of the analysis is to bound all terms (A, B, C, and D) in (3.1). This part is highly problem and algorithm dependent since the terms contain loss estimates and adaptive learning rate terms. In the next part, we bound these terms for the previously described Exp3 algorithm with implicit exploration and comment on some general approaches to bounding these terms.

Bounding term B of (3.1). Since we use the implicit exploration, our loss estimates are negatively biased. This means that $\mathbb{E}[\hat{\ell}_{t,j}]$ can be easily upper-bounded by $\ell_{t,j}$.

This gives us the following lower bound

$$-\mathbb{E}\left[\widehat{L}_{T,j}\right] = -\mathbb{E}\left[\sum_{t=1}^{T}\hat{\ell}_{t,j}\right] \geq -\mathbb{E}\left[\sum_{t=1}^{T}\ell_{t,j}\right] = -\mathbb{E}\Big[L_{T,j}\Big].$$

Bounding term A of (3.1). As we mentioned in the previous paragraph, we usually aim for negatively biased loss estimates. This means that we can not bound term A as easily as term B since it is biased to the other direction. Therefore, we have to use the definition of loss estimates to specify the amplitude of the bias more precisely and show that it does not effect final regret bound. In the case of basic Exp3 with implicit exploration we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\,\middle|\,\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{\ell_{t,i}}{p_{t,i}+\gamma_t}\mathbb{1}\{i=I_t\}\,\middle|\,\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{(p_{t,i}+\gamma_t-\gamma_t)\ell_{t,i}}{p_{t,i}+\gamma_t}\,\middle|\,\mathcal{F}_{t-1}\right] \\
&\geq \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t\sum_{i=1}^{N}\frac{p_{t,i}}{p_{t,i}+\gamma_t}\,\middle|\,\mathcal{F}_{t-1}\right]
\end{aligned}$$

where $\mathcal{F}_{t-1}$ is a history up to the beginning of round t. We also used that $\mathbb{E}\left[\mathbb{1}\{i=I_t\}\,|\,\mathcal{F}_{t-1}\right] = p_{t,i}$ and the fact that $\ell_{t,i} \in [0,1]$. Summing over time and taking an expectation we get

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] \geq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] - \mathbb{E}\left[\sum_{t=1}^{T}\gamma_t Q_t^{\text{Exp3}}\right]$$

where Q_t^{Exp3} is defined as

$$Q_t^{\text{Exp3}} = \sum_{i=1}^{N}\frac{p_{t,i}}{p_{t,i}+\gamma_t}.$$

Note that the first term corresponds to the expected loss of the learner at the end of the game and the second term corresponds to the bias of the learner.

Bounding term D of (3.1). To bound this term we also use the fact that $\mathbb{E}\left[\mathbb{1}\{i = I_t\} \middle| \mathcal{F}_{t-1}\right] = p_{t,i}$ together with $\ell_{t,i} \in [0,1]$. This gives us

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2 \middle| \mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{\ell_{t,i}^2}{(p_{t,i}+\gamma_t)^2}\mathbb{1}\{i = I_t\} \middle| \mathcal{F}_{t-1}\right] \\ &\leq \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{p_{t,i}}{(p_{t,i}+\gamma_t)^2} \middle| \mathcal{F}_{t-1}\right] \\ &\leq \mathbb{E}\left[\sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i}+\gamma_t} \middle| \mathcal{F}_{t-1}\right].\end{aligned}$$

Summing over time and taking expectation we get

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}Q_t^{\text{Exp3}}\right].$$

Bounding term C of (3.1). This term depends only on the definition of the learning rate. In the next part, we choose the value of in order to optimize the regret bound and the final bound can be obtained simply setting learning rate to this value.

Putting everything together we obtain the following bound

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] - \mathbb{E}\left[\sum_{t=1}^{T}\ell_{t,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\left(\gamma_t + \frac{\eta_t}{2}\right)Q_t^{\text{Exp3}}\right].$$

Choosing the best arm j, the left-hand side of the bound is the same as the definition of the regret and thus, we have

$$R_T \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\left(\gamma_t + \frac{\eta_t}{2}\right)Q_t^{\text{Exp3}}\right] \tag{3.2}$$

The next step of the analysis is to specify γ_t and η_t. In order to optimize the bound, we need to specify the constants so that two terms on the right-hand side are of the same order. To deal with the adaptive learning rate in the last term, we use the following lemma

Lemma 21 (Lemma 3.5 of Auer et al., 2002c)**.** *Let b_1, b_2, ..., b_T be non-negative real numbers. Then*

$$\sum_{t=1}^{T} \frac{b_t}{\sqrt{\sum_{s=1}^{t} b_s}} \leq 2\sqrt{\sum_{t=1}^{T} b_t}.$$

Proof. The proof is based on the inequality $x/2 \leq 1 - \sqrt{1-x}$ for $x \leq 1$. Setting $x = b_t / \sum_{s=1}^{t} b_s$ and multiplying both sides of the inequality by $\sqrt{\sum_{s=1}^{t} b_s}$ we get

$$\frac{b_t}{2\sqrt{\sum_{s=1}^{t} b_t}} \leq \sqrt{\sum_{s=1}^{t} b_s} - \sqrt{\sum_{s=1}^{t-1} b_s}.$$

The proof is concluded by summing over t. □

Remark 8. *A usual approach to design algorithms with anytime guarantees is by using doubling trick. However, this comes with the price of a* log *factor in regret bounds. Therefore, we have a different approach. We use adaptive learning rates in all algorithms presented in this chapter which enables us to obtain anytime guarantees.*

Using $\gamma_t = \eta_t/2$ and setting

$$\eta_t = \sqrt{\frac{\log N}{2N + 2\sum_{s=1}^{t-1} Q_t^{\text{Exp3}}}}$$

we make both terms of the bound roughly the same. The first term gives us

$$\mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] \leq \sqrt{2(\log N)\left(N + \sum_{t=1}^{T} Q_t^{\text{Exp3}}\right)}.$$

Using the definition of γ_t and η_t together with the fact that $Q_t^{\text{Exp3}} \leq N$ and Lemma 21 for $b_t = Q_t^{\text{Exp3}}$, the second term can be bounded as

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\gamma_t + \frac{\eta_t}{2}\right) Q_t^{\text{Exp3}}\right] \leq \sqrt{2(\log N)\left(N + \sum_{t=1}^{T} Q_t^{\text{Exp3}}\right)}.$$

Using these bounds we get

$$R_T \leq 2\sqrt{2(\log N)\left(N + \sum_{t=1}^{T} Q_t^{\textsc{Exp3}}\right)}.$$

The last step is to bound $Q_t^{\textsc{Exp3}}$. In fact, it is very simple since every term in the definition of $Q^{\textsc{Exp3}}$ is at most 1 which gives us simple bound $Q_t^{\textsc{Exp3}} \leq N$ and the following theorem.

Theorem 6. *Using adaptive learning rate and implicit exploration, the regret of* Exp3 *is bounded as*

$$R_T \leq 2\sqrt{2N(T+1)\log N} = \widetilde{\mathcal{O}}(\sqrt{NT}).$$

Even though we could simplify the proof by using N instead of $Q^{\textsc{Exp3}}$, we later show that a quantity similar to $Q^{\textsc{Exp3}}$ appears in several regret bounds. Moreover, we are usually able to show tighter bounds on this quantity, using the structure of the problem and the definition of loss estimates.

2 Adversarial bandits with adversarial side observations

In this section, we start by the most studied setting fitting into our framework. Namely, the multi-armed bandit problem with adversarial side observations [Mannor and Shamir, 2011, Alon et al., 2013, Kocák et al., 2014a, Alon et al., 2015] where the observability graph is chosen by an adversary and the side observations are revealed according to this graph.

As we mentioned in Chapter 1, the downside of the first algorithms for this setting is the fact that they need an access to observability graph before playing an action and use a mixing with non-trivial exploration distribution. This results in computationally inefficient algorithms. The first computationally efficient algorithm was Exp3-IX [Kocák et al., 2014a] followed by Exp3.G [Alon et al., 2015]. These algorithms approach exploration in a different way, Exp3-IX uses implicit exploration described in the previous section while Exp3.G uses mixing.

In what follows, we describe the setting in more details and present EXP3-IX algorithm.

2.1 Side-observation setting with adversarial graphs

The problem we consider is defined as follows. In each round $t \in [T]$, the environment assigns a loss vector $\boldsymbol{\ell}_t \in [0,1]^N$ for N actions and also selects an observation system described by the directed graph G_t. Then, based on its previous observations (and likely some external source of randomness) the learner selects action I_t and subsequently incurs and observes loss ℓ_{t,I_t}. Furthermore, the learner also observes the losses $\ell_{t,j}$ for all j such that $(I_t \to j) \in G_t$, denoted by the indicator $O_{t,i}$. Let $\mathcal{F}_{t-1} = \sigma(I_{t-1}, \dots, I_1)$ capture the interaction history up to time t. As usual in online settings Cesa-Bianchi and Lugosi [2006], the performance is measured in terms of (total expected) regret, which is the difference between a total loss received and the total loss of the best single action chosen in hindsight,

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right],$$

where the expectation integrates over the random choices made by the learning algorithm. The usual approach to the problem is by using an algorithm based on EXP3 with mixing

$$\mathbb{P}\left[I_t = i \,|\mathcal{F}_{t-1}\right] = (1-\gamma)p_{t,i} + \gamma\mu_{t,i} = (1-\gamma)\frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}} + \gamma\mu_{t,i},$$

where $\gamma \in (0,1)$ is parameter of the algorithm and $\boldsymbol{\mu}_t$ is an *exploration distribution*. The loss estimates incorporate all the side observation and are defined as

$$\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{o_{t,i}} \mathbb{1}\left\{(I_t \to i) \in G_t\right\} \quad \text{where} \quad o_{t,i} = \mathbb{E}\left[O_{t,i} \,|\mathcal{F}_{t-1}\right] = \mathbb{P}\left[(I_t \to i) \in G_t \,|\mathcal{F}_{t-1}\right],$$

for each $i \in [N]$. These loss estimates are then used to update the weights for all i as

$$w_{t+1,i} = w_{t,i} e^{-\gamma \hat{\ell}_{t,i}}.$$

It is easy to see that these loss estimates $\hat{\ell}_{t,i}$ are unbiased estimates of the true losses whenever $p_{t,i} > 0$ holds for all i. The tricky part is the definition of $\boldsymbol{\mu}_t$. We take a different approach to the problem and use implicit exploration (Section 1.3) instead of mixing.

2.2 Efficient learning by implicit exploration

In this section, we propose the simplest exploration scheme imaginable, which consists of *merely pretending to explore*. Precisely, we simply sample our action I_t from the distribution defined as

$$\mathbb{P}\left[I_t = i \,|\mathcal{F}_{t-1}\right] = p_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}}, \tag{3.3}$$

without explicitly mixing with any exploration distribution. Our key trick is to use implicit exploration to define the loss estimates for all arms i as

$$\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}\left\{(I_t \to i) \in G_t\right\},$$

where $\gamma_t > 0$ is a parameter of our algorithm. It is easy to check that $\hat{\ell}_{t,i}$ is a *biased* estimate of $\ell_{t,i}$. The nature of this bias, however, is very special. First, observe that $\hat{\ell}_{t,i}$ is an *optimistic* estimate of $\ell_{t,i}$ in the sense that $\mathbb{E}\left[\hat{\ell}_{t,i} \,|\mathcal{F}_{t-1}\right] \leq \ell_{t,i}$. That is, our bias always ensures that, on expectation, we underestimate the loss of any fixed arm i. Even more importantly, our loss estimates also satisfy

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i} \,\middle|\, \mathcal{F}_{t-1}\right] &= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} + \sum_{i=1}^{N} p_{t,i}\ell_{t,i}\left(\frac{o_{t,i}}{o_{t,i}+\gamma_t} - 1\right) \\ &= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t \sum_{i=1}^{N} \frac{p_{t,i}\ell_{t,i}}{o_{t,i}+\gamma_t},\end{aligned} \tag{3.4}$$

that is, the bias of the estimated losses *suffered by our algorithm* is directly controlled by γ_t. As we will see in the analysis, it is sufficient to control the bias of our own estimated performance as long as we can guarantee that the loss estimates associated with any fixed arm are optimistic—which is precisely what we have. Note that this slight modification ensures that the denominator of $\hat{\ell}_{t,i}$ is lower bounded by $p_{t,i} + \gamma_t$,

which is a very similar property as the one achieved by the exploration scheme used by EXP3-DOM. In fact, explicit and implicit explorations can both be regarded as two different approaches for bias-variance tradeoff: while explicit exploration biases the *sampling distribution* of I_t to reduce the variance of the loss estimates, implicit exploration achieves the same result by biasing *the loss estimates themselves*.

2.3 EXP3-IX algorithm and theoretical guarantees

We define our algorithm EXP3-IX as a variant of EXP3 using the IX loss estimates. One of the twists is that EXP3-IX is actually based on the adaptive learning-rate variant of EXP3 proposed by Auer et al. [2002c], which avoids the necessity of prior knowledge of the observability graphs in order to set a proper learning rate. This algorithm is defined by setting $\widehat{L}_{t-1,i} = \sum_{s=1}^{t-1} \hat{\ell}_{s,i}$ and for all $i \in [N]$ computing the weights as

$$w_{t,i} = \frac{1}{N} e^{-\eta_t \widehat{L}_{t-1,i}}.$$

These weights are then used to construct the sampling distribution of I_t as defined in (3.3). The resulting EXP3-IX algorithm is shown as Algorithm 5.

Algorithm 5 EXP3-IX

Input: Set of actions $\mathcal{S} = [N]$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  $w_{t,i} = (1/N) \exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for $i \in [N]$
  An adversary privately chooses losses $\ell_{t,i}$ for $i \in [N]$ and generates a graph G_t
  $W_t = \sum_{i=1}^{N} w_{t,i}$
  $p_{t,i} = w_{t,i}/W_t$
  Choose $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,N})$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for $(I_t \to i) \in G_t$
  $o_{t,i} = \sum_{(j \to i) \in G_t} p_{t,j}$ for $i \in [N]$
  $\hat{\ell}_{t,i} = \frac{\ell_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}}$ for $i \in [N]$
end for

Our analysis follows the footsteps of Auer et al. [2002b] and Györfi and Ottucsák [2007], who provide an improved analysis of the adaptive learning-rate rule proposed

by Auer et al. [2002c]. However, a technical subtlety will force us to proceed a little differently than these standard proofs: for achieving the tightest possible bounds and the most efficient algorithm, we need to tune our learning rates according to some random quantities that depend on the performance of Exp3-IX. In fact, the key quantities in our analysis are the terms

$$Q_t^{\mathrm{IX}} = Q(1, 0, \gamma_t) = \sum_{i=1}^{N} \frac{p_{t,i}}{o_{t,i} + \gamma_t},$$

which depend on the interaction history $\mathcal{F}_{t-1}$ for all t. Our theorem below gives the performance guarantee for Exp3-IX using a parameter setting adaptive to the values of Q_t^{IX}.

Theorem 7. *Setting* $\eta_t = \sqrt{(\log N)/2 \left(N + \sum_{s=1}^{t-1} Q_s^{\mathrm{IX}}\right)}$ *and* $= \gamma_t = \eta_t/2$*, the regret of* Exp3-IX *satisfies*

$$R_T \leq \mathbb{E}\left[\sqrt{8(\log N)\left(N + \sum_{t=1}^{T} Q_t^{\mathrm{IX}}\right)}\right]. \tag{3.5}$$

Full proof of the theorem is provided later in section 6.1. The next step is to connect this regret bound to the observability graph and bound Q_t^{IX} by a deterministic quantity depending on the graph. However, similar quantities like Q^{IX} appear in our other setting later in this chapter. Therefore, we generalize Q^{IX} and bound this more general version. Later we reuse this result in other settings. Let us generalize Q^{IX}.

Definition 9. *Let* m *be a positive integer,* c *be a positive constant,* δ *be a non-negative constant, and* G *be an oriented graph with weight* $s_{j,i}$ *on the edge from* j *to* i *(*$s_{j,i} = 0$ *if there is no edge from* j *to* i*). Then*

$$Q(m, \delta, c) = \sum_{i=1}^{N} \frac{p_i}{\frac{1}{m} p_i + \frac{1}{m} \sum_{j \neq i} p_j s_{j,i}^{1+\delta} + c}.$$

Note that the definition of Q_t^{IX} is a special case the definition of $Q(m, \delta, c)$ for $m = 1$, $\delta = 0$, and $c = \gamma_t$. Before bounding $Q(m, \delta, c)$, we need one more definition.

Definition 10. *The independence number* α *of graph* G *is the size of the largest independence set; set of nodes without edges connecting any two of the nodes. Therefore, the independence number of the graph is 6.*

The next figure shows a small example of the graphs with its largest independence set of size 6.

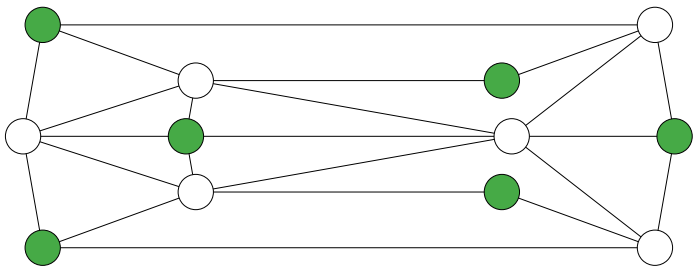


Figure 3.2: Six green nodes form the largest independence set of the graph; there is no edge between any two of the green nodes.

Now we are ready to bound $Q(m, \delta, c)$. The following bound is a generalization of Lemma 13 of Alon et al. [2013].

Lemma 22. *Let G be a directed weighted graph with vertex set $V = \{1, \ldots, N\}$. Let $s_{j,i}$ be a weight corresponding to the edge from j to i. Let $\alpha(\varepsilon)$ be the independence number of G after removing all the edges with weights smaller than ε and $p_1, \ldots, p_N$ are numbers from $[0, 1]$ such that $\sum_{i=1}^{N} p_i \leq m$. Then for any $\varepsilon \in [0, 1]$, positive constant c, and non-negative constant δ we have*

$$Q(m, \delta, c) \leq 2m \frac{\alpha(\varepsilon)}{\varepsilon^{1+\delta}} \left[1 + \log\left(1 + \frac{N^2 \varepsilon^{1+\delta} + 2Nc}{c\alpha(\varepsilon)}\right)\right]$$

Remark 9. *We introduced extra constant δ in the previous lemma. The role of this constant is to unify analyses of the algorithms in this chapter and it also enables us to tune bounds more precisely.*

This lemma gives us a way to construct a deterministic bound on $Q(m, \delta, c)$. Moreover, it generalizes Lemma 13 of Alon et al. [2013] in several ways. First, we no longer require $(p_i)_{i=1}^{N}$ to be a probability distribution. Instead we assume that $\sum_{i=1}^{N} p_i \leq m$. This enables us to extend our framework to the combinatorial case. Second, we generalize this lemma to the case of weighted graphs. Last, we allow different powers of edge weights $s_{j,i}$. Later, we show that these generalizations are crucial for some of the extensions proposed later in this chapter.

Proof. The proof relies on the following two statements borrowed from Alon et al. [2013]. The first statement is an application of Turán's theorem on the complemen-

tary graph. This gives us a standard graph theoretical lemma connecting indegrees of vertices to the independence number of a graph.

Lemma 23 (Lemma 10 of Alon et al. [2013]). *Let G be a directed graph, with $V = \{1, \dots, N\}$. Let d_i^- be the indegree of the node i and $\alpha = \alpha(G)$ be the independence number of G. Then*

$$\sum_{i=1}^{N} \frac{1}{1 + d_i^-} \leq 2\alpha \log\left(1 + \frac{N}{\alpha}\right).$$

For the second statement, we use simple algebraic identities and inequalities.

Lemma 24 (Lemma 12 of Alon et al. [2013]). *If $a, b \geq 0$ and $a + b \geq B > A > 0$, then*

$$\frac{a}{a + b - A} \leq \frac{a}{a + b} + \frac{A}{B - A}$$

Proof.

$$\frac{a}{a + b - A} - \frac{a}{a + b} = \frac{aA}{(a + b)(a + b - A)} \leq \frac{A}{a + b - A} \leq \frac{A}{B - A}$$

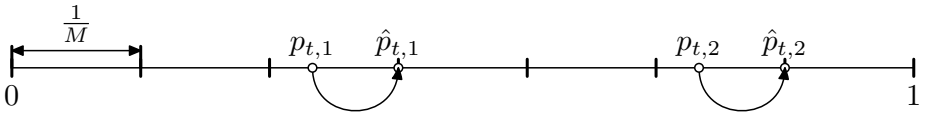
□

We are now ready to prove Lemma 22. Our proof is obtained as a generalization of the proof of Lemma 13 by Alon et al. [2013]. Let us recall the definition of $Q(m, \delta, c)$.

$$Q(m, \delta, c) = \sum_{i=1}^{N} \frac{p_i}{\frac{1}{m} p_i + \frac{1}{m} \sum_{j \neq i} p_j s_{j,i}^{1+\delta} + c}$$

We begin by constructing a discretization $\hat{p}_i$ of the values p_i, for every $i \in [N]$. The discretization satisfies $\hat{p}_i = k/M$ for some integer k such that $\hat{p}_i - 1/M < p_i \leq \hat{p}_i$ where M depends on $Q(m, \delta, c)$ and is defined as

$$M = \left\lceil \frac{N^2 \varepsilon^{1+\delta}}{mc} \right\rceil.$$



This allows us to upper bound $Q(m, \delta, c)$ as

$$\begin{aligned} Q(m,\delta,c) &\leq m \sum_{i=1}^{N} \frac{p_i}{\varepsilon^{1+\delta} p_i + \sum_{j \neq i} p_j \varepsilon^{1+\delta} \mathbb{1}\{s_{j,i} \geq \varepsilon\} + mc} \\ &\leq \frac{m}{\varepsilon^{1+\delta}} \sum_{i=1}^{N} \frac{\hat{p}_i}{\hat{p}_i + \sum_{j \neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\} + \frac{mc}{\varepsilon^{1+\delta}} - \frac{N}{M}} \\ &\leq \frac{m}{\varepsilon^{1+\delta}} \sum_{i=1}^{N} \left(\frac{\hat{p}_i}{\hat{p}_i + \sum_{j \neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\} + \frac{mc}{\varepsilon^{1+\delta}}} + \frac{\frac{N}{M}}{\frac{mc}{\varepsilon^{1+\delta}} - \frac{N}{M}} \right). \end{aligned}$$

In the last step, we used Lemma 24 with $a = \hat{p}_i$, $b = \sum_{j \neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\} + mc/\varepsilon^{1+\delta}$, $A = N/M$, and $B = mc/\varepsilon^{1+\delta}$. Using the definition of M, we can easily bound the second fraction in the previous expression as

$$\frac{\frac{N}{M}}{\frac{mc}{\varepsilon^{1+\delta}} - \frac{N}{M}} = \frac{N\varepsilon^{1+\delta}}{Mmc - \varepsilon^{1+\delta}N} \leq \frac{N\varepsilon^{1+\delta}}{\varepsilon^{1+\delta}N^2 - \varepsilon^{1+\delta}N} = \frac{N\varepsilon^{1+\delta}}{\varepsilon^{1+\delta}N(N-1)} \leq \frac{2}{N}.$$

Using this inequality, we can continue bounding $Q(m, \delta, c)$ as

$$\begin{aligned} Q(m,\delta,c) &\leq \frac{m}{\varepsilon^{1+\delta}} \sum_{i=1}^{N} \left(\frac{\hat{p}_i}{\hat{p}_i + \sum_{j \neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\}} + \frac{2}{N} \right) \\ &= \frac{m}{\varepsilon^{1+\delta}} \left(2 + \sum_{i=1}^{N} \frac{\hat{p}_i}{\hat{p}_i + \sum_{j \neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\}} \right). \end{aligned}$$

It remains to find a suitable upper bound for the last sum.

The last part of the proof is to construct a graph G' from our original graph G by deleting all the edges with weights smaller than ε (thresholding), removing the edge orientation, and replacing each node i of G by a clique C_i with $M\hat{p}_i$ nodes. In this expanded graph, we connect all vertices in clique C_i with all vertices in C_j if and only if there is an edge from i to j in thresholded G. Note that our new graph G' has the same thresholded independence number $\alpha(\varepsilon)$ as the original graph G after

thresholding. Also observe that the indegree $\hat{d}_k^-$ of a node k in clique C_i is equal to $M\hat{p}_i - 1 + \sum_{j\neq i} M\hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\}$. Therefore, the last sum can be rewritten as

$$\begin{aligned}\sum_{i=1}^{N} \frac{\hat{p}_i}{\hat{p}_i + \sum_{j\neq i} \hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\}} &= \sum_{i=1}^{N} \frac{M\hat{p}_i}{M\hat{p}_i + \sum_{j\neq i} M\hat{p}_j \mathbb{1}\{s_{j,i} \geq \varepsilon\}} \\ &= \sum_{i=1}^{N} \sum_{k\in C_i} \frac{1}{1+\hat{d}_k^-}\end{aligned}$$

which in turn can be bounded using Lemma 23 by

$$2\alpha(\varepsilon)\log\left(1 + \frac{\sum_{i=1}^{N} M\hat{p}_i}{\alpha(\varepsilon)}\right) \leq 2\alpha(\varepsilon)\log\left(1 + \frac{mM+N}{\alpha(\varepsilon)}\right).$$

Using this bound together with $\alpha(\varepsilon) \geq 1$ and the definition of M, we get

$$Q(m,\delta,c) \leq 2m\,\frac{\alpha(\varepsilon)}{\varepsilon^{1+\delta}}\left[1 + \log\left(1 + \frac{N^2\varepsilon^{1+\delta} + 2Nc}{c\alpha(\varepsilon)}\right)\right]$$

as advertised. □

Now, we use Lemma 22, with $m = 1$, $\delta = 0$, and $c = \gamma_t$, to get the final result for the Exp3-IX algorithm.

Corollary 2. *The regret of* Exp3-IX *satisfies*

$$R_T \leq \sqrt{8(\log N)\left(N + 2\sum_{t=1}^{T} H_t\alpha_t\right)},$$

where α_t is the independence number of the graph at time t and

$$H_t = 1 + \log\left(1 + \frac{2N + N^2\sqrt{NT/\log(N)}}{\alpha_t}\right) = \mathcal{O}(\log(TN)).$$

Note that the setting of the Exp3-IX algorithm is with the perfect side observations i.e. all the weights are equal to 1. This means that α_t does not depend on the thresholding parameter ε since setting ε to any value in $[0, 1]$ does not change the value of $\alpha_t(\varepsilon)$.

Proof. Using Lemma 22, with $m = 1$, $\delta = 0$, $c = \gamma_t$, and setting ε to 1, we can show that H_t can be bounded as

$$\begin{aligned} H_t &= 1 + \log\left(1 + \frac{N^2/\gamma_t + 2N}{\alpha_t}\right) \\ &\leq 1 + \log\left(1 + \frac{2N + N^2\sqrt{NT/\log(N)}}{\alpha_t}\right) \\ &= \mathcal{O}(\log(TN)). \end{aligned}$$

We used the fact that Q_t^{IX} can be trivially bounded by N to bound γ_t. □

3 Adversarial multi-armed bandit problem with stochastic side observations

The main drawback of the basic bandits with side observations [Mannor and Shamir, 2011], studied in Section 2, is that the learner requires the environment to reveal a substantial part of a graph, at least after the action is taken [Cohen et al., 2016]. Specifically, the learner requires the knowledge of the *second neighborhood* (the set of neighbors of the neighbors) of the chosen action in order to update their internal loss estimates. On the other hand, the algorithms are able to deal with any graph structures.

The main contribution of this section is a learning algorithm that, unlike previous solutions, does *not require the knowledge of the exact graph* underlying the observations, beyond knowing from which nodes the side observations came from. Relaxing this assumption, however, has to come with a price: As the very recent results of Cohen, Hazan, and Koren [2016] show, achieving nontrivial advantages from side observations may be impossible without perfectly known side-observation graphs when an adversary is allowed to pick *both* the losses and the side-observation graphs. On the positive side, Cohen et al. offer efficient algorithms achieving strong improvements over the standard regret guarantees under the assumption that the losses are generated in an i.i.d. fashion and the graphs may be generated adversarially. Complementing these results, we consider the case of adversarial losses and make the assumption that the side-observation graph in round t is generated from an *Erdős–Rényi* model with an *unknown* and *time-dependent* parameter r_t (Figure 3.3). The main challenge for the learner is then the necessity to exploit the side observations

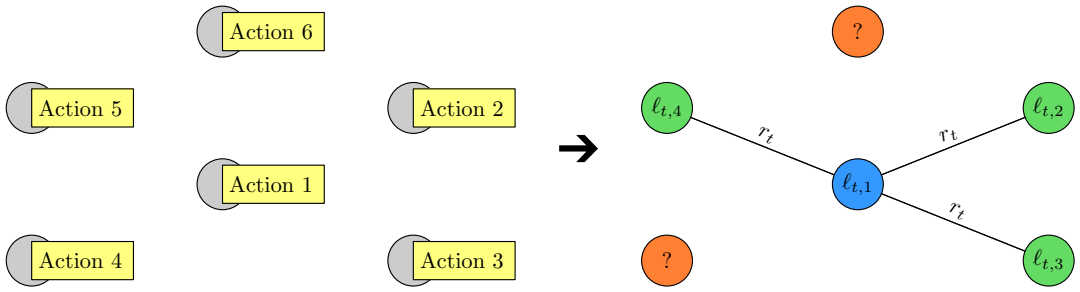


Figure 3.3: The learner picks an action (blue node) and observes losses of other actions with probability r_t

despite not knowing the sequence (r_t). It is easy to see that this model can be equivalently understood as each non-chosen arm revealing its loss with probability r_t, independently of all other observations. That said, we still find it useful to think of the side observations as being generated from an Erdős–Rényi model, as it allows direct comparisons with the related literature. In particular, the case of learning with Erdős–Rényi side-observation graphs was considered before by Alon et al. [2013]: Given *full access* to the underlying graph structure, their algorithm Exp3-SET can be shown to guarantee a regret bound of $\mathcal{O}\big(\sqrt{\sum_t (1/r_t)(1-(1-r_t)^N)\log N}\big)$. While the assumption of having full access to the graph be dropped relatively easily in this particular case, exact knowledge of r_t seems to be crucial for constructing reliable loss estimates and use them to guide the choice of action in each round.

It turns out that the problem of estimating r_t while striving to perform efficiently is, in fact, a major difficulty in our setting. Indeed, as we allow r_t to change arbitrarily between each round, we cannot rely on any past observations to construct well-concentrated estimates of these parameters. That is, the main challenge is estimating r_t from only a handful of samples. The core technical tool underlying our approach is a direct estimation procedure for the losses that does not estimate r_t explicitly.

Armed with this estimation procedure, we propose a learning algorithm called Exp3-Res that guarantees a regret of $\mathcal{O}(\sqrt{\sum_t (1/r_t)\log N})$, provided that the condition $r_t \geq \log T/(2N-2)$ holds for all rounds t. This assumption essentially corresponds to requiring that, with high probability, at least 1 side observation is produced in every round, or, in other words, the side-observation graphs encountered are all *non-empty*. Notice that for the assumed range of r_t's, our regret bound improves upon the standard regret bound of Exp3, which is of $\mathcal{O}(\sqrt{NT\log N})$. It is easy to see that when r_t becomes smaller than $1/N$, side observations become unreliable and the bound of Exp3 cannot be improved. That is, if our assumption cannot be verified

a priori, then ignoring all side observations and using the EXP3 algorithm of Auer et al. [2002b] instead can yield a better performance. On the other hand, given that our assumption holds, our bounds cannot be significantly improved as suggested by the lower bound of $\Omega(\sqrt{T/r})$ proved for a static r by Alon et al. [2013].

Many other partial-information settings have been studied in previous work. One of the simplest of these settings is the label-efficient prediction game considered by Cesa-Bianchi et al. [2005], where the learner can observe either losses of all the actions or none of them, not even the loss of the chosen action. This observation can be queried by the learner at most an $\varepsilon < 1$ fraction of the total number of rounds, which means no losses are observed in the remaining rounds. An even more restricted information setting, label-efficient bandit feedback was considered by Allenberg et al. [2006], where the learner can only query the loss of the chosen action, instead of all losses (see also Audibert and Bubeck, 2010). Algorithms for these two settings have regret of $\widetilde{\mathcal{O}}(\sqrt{T/\varepsilon})$ and $\widetilde{\mathcal{O}}(\sqrt{NT/\varepsilon})$, respectively. While these bounds may appear very similar to ours, notice that our setting offers a more intricate (and, for some problems, more realistic) feedback scheme, which also turns out to be much more challenging to exploit. In another related setting, Seldin et al. [2014] consider M side observations that the learner can proactively choose in each round without limitations. Seldin et al. deliver an algorithm with regret of $\widetilde{\mathcal{O}}(\sqrt{(N/M)T})$, also proving that choosing M observations uniformly at random is minimax optimal; given this sampling scheme, it is not even necessary to observe the loss of the chosen action. Their result is comparable to ours and the result by Alon et al. [2013] for Erdős–Rényi observation graphs with parameter $r = M/N$. However, Seldin et al. also assume that M is known, which obviates the need for estimating r.

In this section, we assume that, just like the observation probabilities, the losses are *adversarial*, that is, they can change at each time step without restrictions. Learning with side observations and stochastic losses was studied by Caron et al. [2012] and Buccapatnam et al. [2014]. While this is an easier setting that the adversarial one, the authors assumed, in both cases, that the graphs have to be known in advance. Recently, Carpentier and Valko [2016] studied another stochastic setting where the graph is also not known in advance, however, their setting considers different feedback and loss structure (influence maximization) which differs from the side-observation setting.

3.1 Side-observation setting with stochastic graphs

We now formalize our learning problem. We consider a sequential interaction scheme between a learner and an environment, where the following steps are repeated in every round $t = 1, 2, \ldots, T$:

1. The environment chooses $r_t \in [0, 1]$ and a loss function over the arms, with $\ell_{t,i}$ being the loss associated with arm $i \in [N] \stackrel{\text{def}}{=} \{1, 2, \ldots, N\}$ at time t.
2. Based on its previous observations (and possibly some randomness), the learner draws an arm $I_t \in [N]$.
3. The learner suffers loss ℓ_{t,I_t}.
4. For all $i \neq I_t$, $O_{t,i}$ is independently drawn from a Bernoulli distribution with mean r_t. Furthermore, O_{t,I_t} is set as 1.
5. For all $i \in [N]$ such that $O_{t,i} = 1$, the learner observes the loss $\ell_{t,i}$.

The goal of the learner is to minimize its total expected loss, or, equivalently, to minimize the *total expected regret* (or, in short, regret) defined as

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} (\ell_{t,I_t} - \ell_{t,i})\right].$$

We will denote the interaction history between the learner and the environment up to the beginning of round t by $\mathcal{F}_{t-1}$. We also define $p_{t,i} = \mathbb{P}\left[I_t = i \middle| \mathcal{F}_{t-1}\right]$.

The main challenge in our setting is leveraging side observations *without knowing* r_t. Had we had access to the exact value of r_t, we would be able to define the following estimate of $\ell_{t,i}$:

$$\hat{\ell}^{\star}_{t,i} = \frac{O_{t,i}\ell_{t,i}}{p_{t,i} + (1 - p_{t,i})r_t} \tag{3.6}$$

It is easy to see that the loss estimates defined this way are unbiased in the sense that $\mathbb{E}\left[\hat{\ell}_{t,i} \middle| \mathcal{F}_{t-1}\right] = \ell_{t,i}$ for all t and i. It is also straightforward to show that an appropriately tuned instance of the Exp3 algorithm of Auer et al. [2002b] fed with

these loss estimates is guaranteed to achieve a regret of $\mathcal{O}(\sqrt{\sum_t (1/r_t) \log N})$ (see also Seldin et al. 2014).

Then, one might consider a simple algorithm that devotes a number of observations to obtain an estimate $\widehat{r}_t$ of r_t and plug this estimate into (3.6). However, notice that since r_t is allowed to change arbitrarily over time, we can only work with a severely limited sample budget for estimating r_t: only $N-1$ independent observations! Thus, we can obtain only very loose confidence intervals around r_t which translate to even more useless confidence intervals around $\hat{\ell}^\star_{t,i}$.

Below, we describe a simple trick for obtaining loss estimates that have similar properties to the ones defined in (3.6) without requiring exact knowledge or even explicit estimation of r_t. Our procedure is based on the geometric resampling method of Neu and Bartók [2013]. To get an intuition of the method, let us assume that we have access to the independent geometrically distributed random variable $G^\star_{t,i}$ with parameter $o_{t,i} = p_{t,i} + (1 - p_{t,i})r_t$. Then, replacing $1/o_{t,i}$ by $G^\star_{t,i}$ in the definition of $\hat{\ell}^\star_t$ and ensuring that $G^\star_{t,i}$ is independent of $O_{t,i}$, we can obtain an unbiased loss estimate essentially equivalent to $\hat{\ell}^\star_t$.

The challenge posed by this approach is that in our setting, we do not have exact sample access to the geometric random variable $G^\star_{t,i}$. In the next section, we describe our algorithm that is based on replacing $G^\star_{t,i}$ in the above definition by an appropriate surrogate.

3.2 EXP3-RES algorithm and theoretical guarantees

Our algorithm is called EXP3-RES and displayed as Algorithm 6. It is based on the EXP3 algorithm of Auer et al. [2002b] and crucially relies on the construction of a surrogate $G_{t,i}$ of $G^\star_{t,i}$. Throughout this section, we will assume that $r_t \geq \frac{\log T}{2N-2}$, which implies that the probability of having no side observations in round t is of order $1/\sqrt{T}$.

The algorithm is initialized by setting $w_{1,i} = 1/N$ for all $i \in [N]$, and then performing the updates

$$w_{t+1,i} = \frac{1}{N} \exp\left(-\eta_{t+1} \widehat{L}_{t,i}\right) \tag{3.7}$$

after each round t, where $\eta_{t+1} > 0$ is an adaptive learning rate parameter of the

algorithm at round t and $\widehat{L}_{t,i}$ is cumulative sum of the loss estimates $\hat{\ell}_{s,i}$ up to (and including) time t. In round t, the learner draws its action I_t such that $I_t = i$ holds with probability $p_{t,i} \propto w_{t,i}$. To simplify some of the notation below, we introduce the shorthand notations $\mathbb{P}_t[\cdot] = \mathbb{P}[\cdot \,|\, \mathcal{F}_{t-1}]$ and $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \,|\, \mathcal{F}_{t-1}]$.

Algorithm 6 EXP3-RES

Input:
  Set of actions $[N]$.
Initialization:
  $\widehat{L}_{0,i} = 0$ for $i \in [N]$.
Run:
for $t = 1$ **to** T **do**
  $\eta_t = \sqrt{\log N \Big/ \left(N^2 + \sum_{s=1}^{t-1} \sum_{i=1}^{N} p_{s,i} (\hat{\ell}_{s,i})^2\right)}$.
  $w_{t,i} = (1/N) \exp(-\eta_t \widehat{L}_{t-1,i})$ for $i \in [N]$.
  $W_t = \sum_{i=1}^{N} w_{t,i}$.
  $p_{t,i} = w_{t,i} / W_t$.
  Choose $I_t \sim p_t = (p_{t,1}, \ldots, p_{t,N})$.
  Receive the observation set O_t.
  Receive the pairs $\{i, \ell_{t,i}\}$ for all i s.t. $O_{t,i} = 1$.
  Compute $G_{t,i}$ for all $i \in [N]$ using (3.8).
  $\hat{\ell}_{t,i} = \ell_{t,i} O_{t,i} G_{t,i}$ for all $i \in [N]$.
  $\widehat{L}_{t,i} = \widehat{L}_{t-1,i} + \hat{\ell}_{t,i}$ for all $i \in [N]$.
end for

For any fixed t, i, we now describe an efficiently computable surrogate $G_{t,i}$ for the geometrically distributed random variable $G^{\star}_{t,i}$ with parameter $o_{t,i}$ that will be used for constructing our loss estimates. In particular, our strategy will be to construct several independent copies $\left\{O'_{t,i}(k)\right\}$ of $O_{t,i}$ and choosing $G_{t,i}$ as the index k of the first copy with $O'_{t,i}(k) = 1$. It is easy to see that with infinitely many copies, we could exactly recover $G^{\star}_{t,i}$; our actual surrogate is going to be weaker thanks to the smaller sample size. For clarity of notation, we will omit most explicit references to t and i, with the understanding that all calculations need to be independently executed for all pairs t, i.

Let us now describe our mechanism for constructing the copies $\{O'(k)\}$. Since we need independence of $G_{t,i}$ and $O_{t,i}$ for our estimates, we use only side observations from actions $[N] \setminus \{I_t, i\}$. First, let's define σ as a uniform random permutation of $[N] \setminus \{I_t, i\}$. For all $k \in [N-2]$, we define $R(k) = O_{t,\sigma(k)}$. Note that due to the construction, $\{R(k)\}_{k=1}^{N-2}$ is an independent set of Bernoulli random variables

with parameter r_t, independent of $O_{t,i}$. Furthermore, knowing $p_{t,i}$ we can define $P(1), \ldots, P(N-2)$ as pairwise independent Bernoulli random variables with parameter $p_{t,i}$. Using $P(k)$ and $R(k)$ we define the random variable $O'(k)$ as

$$O'(k) = P(k) + (1 - P(k))R(k)$$

for all $k \in [N-2]$. Using independence of all previously defined random variables, it is easy to check that the variables $\{O'(k)\}_{k=1}^{N-2}$ are pairwise independent Bernoulli random variables with expectation $o_{t,i} = p_{t,i} + (1-p_{t,i})r_t$. Now we are ready to define $G_{t,i}$ as

$$G_{t,i} = \min\left\{k \in [N-2] : O(k)' = 1\right\} \cup \left\{N-1\right\}. \tag{3.8}$$

The following lemma states some properties of $G_{t,i}$.

Lemma 25. *For any value of $o_{t,i}$ we have*

$$\begin{aligned}
\mathbb{E}\left[G_{t,i}\right] &= \frac{1}{o_{t,i}} - \frac{1}{o_{t,i}}(1-o_{t,i})^{N-1} \\
\mathbb{E}\left[G_{t,i}^2\right] &= \frac{2-o_{t,i}}{o_{t,i}^2} + \frac{1}{o_{t,i}^2}(1-o_{t,i})^{N-2}\Big(o_{t,i}^2 + o_{t,i} - 2 + 2o_{t,i}(N-2)(o_{t,i}-1)\Big)
\end{aligned}$$

Proof. The proof follows directly from using the definition of $G_{t,i}$ and simplifying the sums

$$\begin{aligned}
\mathbb{E}\left[G_{t,i}\right] &= \sum_{k=1}^{N-2}\left[k o_{t,i}(1-o_{t,i})^{k-1}\right] + (N-1)\left(1-o_{t,i}\right)^{N-2}, \\
\mathbb{E}\left[G_{t,i}^2\right] &= \sum_{k=1}^{N-2}\left[k^2 o_{t,i}(1-o_{t,i})^{k-1}\right] + (N-1)^2\left(1-o_{t,i}\right)^{N-2}.
\end{aligned}$$

□

Using Lemma 25, it is easy to see that $G_{t,i}$ follows a truncated geometric law in the sense that

$$\mathbb{P}\left[G_{t,i} = m\right] = \mathbb{P}\left[\min\left\{G_{t,i}^\star, N-1\right\} = m\right]$$

holds for all $m \in [N-1]$. Using all this notation, we construct an estimate of $\ell_{t,i}$ as

$$\hat{\ell}_{t,i} = G_{t,i} O_{t,i} \ell_{t,i}. \tag{3.9}$$

The rationale underlying this definition of $G_{t,i}$ is rather delicate. First, note that $p_{t,i}$ is deterministic given the history $\mathcal{F}_{t-1}$ and therefore, does not depend on $O_{t,i}$. Second, $O_{t,i}$ is also independent of $O_{t,j}$ for $j \notin \{i, I_t\}$. As a result, $G_{t,i}$ is independent of $O_{t,i}$, and we can use the identity $\mathbb{E}_t[G_{t,i} O_{t,i}] = \mathbb{E}_t[G_{t,i}]\,\mathbb{E}_t[O_{t,i}]$. The next lemma relates the loss estimates (3.9) to the true losses, relying on the observations above and the assumption $r_t \geq \frac{\log T}{2N-2}$.

Lemma 26. *Assume $r_t \geq \frac{\log T}{2N-2}$. Then, for all t and i,*

$$0 \leq \ell_{t,i} - \mathbb{E}_t\left[\hat{\ell}_{t,i}\right] \leq \frac{1}{\sqrt{T}}.$$

Proof. Fix an arbitrary t and i. Using Lemma 25 along with $\mathbb{E}_t[O_{t,i}] = o_{t,i}$ and the independence of $G_{t,i}$ and $O_{t,i}$, we get

$$\mathbb{E}_t\left[\hat{\ell}_{t,i}\right] = \mathbb{E}_t[G_{t,i} O_{t,i} \ell_{t,i}] = \ell_{t,i} - \ell_{t,i}(1 - o_{t,i})^{N-1},$$

which immediately implies the lower bound on $\ell_{t,i} - \mathbb{E}_t\left[\hat{\ell}_{t,i}\right]$. For proving the upper bound, observe that

$$\ell_{t,i}(1 - o_{t,i})^{N-1} \leq (1 - r_t)^{N-1} \leq e^{-r_t(N-1)} \leq \frac{1}{\sqrt{T}}$$

holds by our assumption on r_t, where we used the elementary inequality $1 - x \leq e^x$ that holds for all $x \in \mathbb{R}$. □

The next theorem states our main result concerning Exp3-Res with an adaptive learning rate η_t.

Theorem 8. *Assume that $r_t \geq \frac{\log T}{2N-2}$ holds for all t and set*

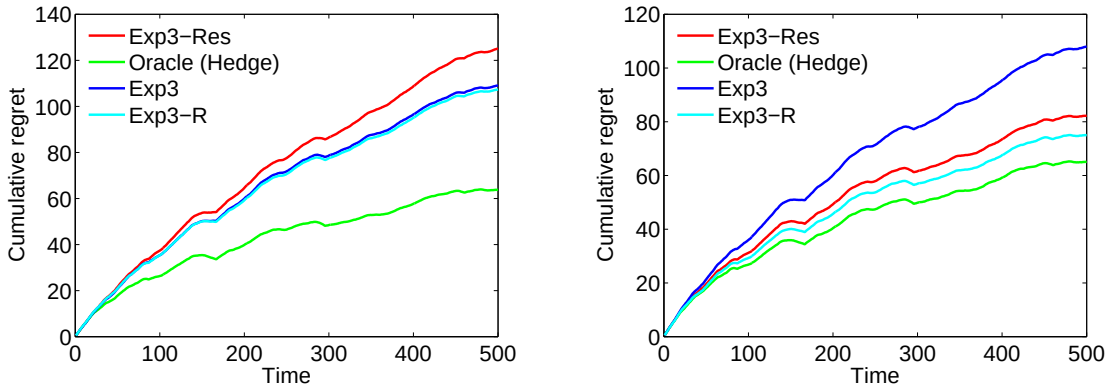
$$\eta_t = \sqrt{\frac{\log N}{N^2 + \sum_{s=1}^{t-1} \sum_{i=1}^{N} p_{s,i} (\hat{\ell}_{s,i})^2}}.$$

Then, the expected regret of Exp3-Res *satisfies*

$$R_T \leq 2\sqrt{\left(N^2 + \sum_{t=1}^{T} \frac{1}{r_t}\right) \log N} + \sqrt{T}.$$

Detailed proof of the theorem can be found in Section 6.2

3.3 Experiments



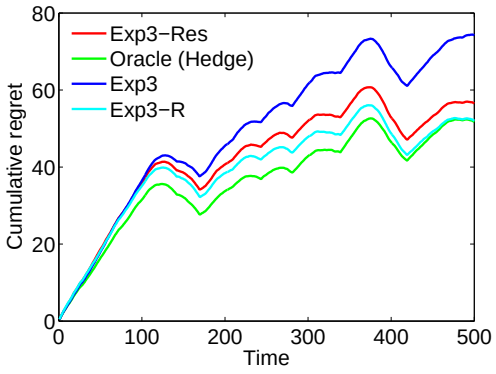
(a) Static sequence $(r_t)_t^T$ where $r_t = 0$ for all t

(b) Static sequence $(r_t)_t^T$ where $r_t = 0.06 \approx \log(T)/(2N-2)$

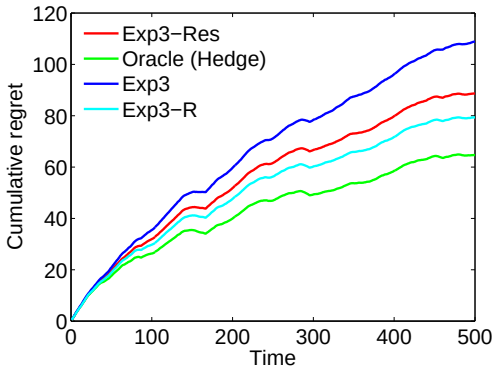
Figure 3.4: Comparison of the algorithms for different amount of side information and fixed value of r_t

In this section, we study the empirical performance of Exp3-Res compared to three other algorithms:

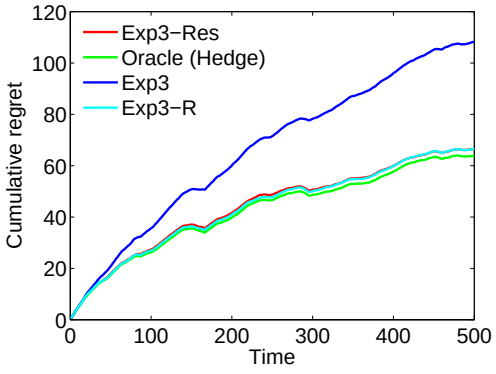
- `Exp3` – a basic adversarial multi-armed bandit algorithm which uses only loss observations of chosen arms and discards all side observations.
- Oracle – full-information algorithm with access to losses of every action in every time step, regardless of the value of r_t. Our particular choice is Hedge [Littlestone and Warmuth, 1994, Freund and Schapire, 1997].
- Exp3-R – a variant of the Exp3-Res algorithm with access to the sequence $(r_t)_t^T$, using (3.6) to construct unbiased loss estimate instead of using geometric resampling.



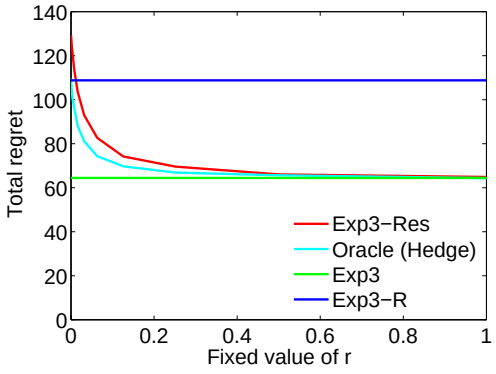
(a) Changing sequence $(r_t)_t^T$ with uniformly distributed r_t on $[0, 0.2]$



(b) Sequence $(r_t)_t^T$ generated as a random walk on $[0, 0.1]$



(c) Sequence $(r_t)_t^T$ generated as a random walk on $[0, 1]$



(d) Total regret for different values of static $(r_t)_t^T$

Figure 3.5: Comparison of the algorithms for different side information sequences (different sequences $(r_t)_t^T$)

The most interesting parameter of our experiment is the sequence (r_t), since it controls the amount of side observation presented to the learner. In order to show that EXP3-RES can effectively make use of the additional information provided by the environment, we designed several sequences (r_t) with different amounts of side observation provided to the learner. In the case of small r_t-s, the problem is almost as difficult as the multi-armed bandit problem. On the other hand, in the case of large r_t-s, the problem is almost as easy as the full-information problem. Therefore, we expect that the performance of EXP3-RES will interpolate between the performance of the EXP3-R and ORACLE algorithms depending on the values of the r_t-s. In the next section, we validate this claim empirically.

To ensure sufficient challenge for the algorithms, we have generated a sequence of losses as a random walk for each arm with independent increments uniformly distributed on $[-0.1, 0.1]$ while enforcing the random walks to stay within $[0, 1]$ by setting the value of a random walk to 0 or 1, respectively, if the random walk gets outside the boundaries. The loss sequence is fixed through all of the experiments to demonstrate the impact of the sequence $(r_t)_t^T$ on the regret of algorithms. We have observed qualitatively similar behavior for other loss sequences.

We fix the number of arms in all of the experiments as 50, and the time horizon as 500. Every curve represents an average of 100 runs.

We performed experiments on many different loss sequences and sequences of r_t-s. Since the results are essentially the same for all the different sequences, we included just the results for one loss sequence with different sequences of r_t-s. In the case of $r_t \geq \log(T)/(2N-2)$, the case of a high probability of having some side observation, the performance of the algorithm Exp3-Res proposed in the present paper is comparable to the performance of the idealistic Exp3-R which knows the exact value of r_t in every time step. Moreover, if the average r_t is close to 1, the performance of the proposed algorithm is close to the performance of Oracle which observes all the losses. If the average r_t is close to zero, the performance of the algorithm is a little bit worse that the performance of basic Exp3. This is also supported by the theory since our algorithm is not able to construct reliable estimates in the case of small r_t-s.

4 Adversarial multi-armed bandit problem with noisy side observations

In the previous sections, we studied settings which assumed that the side observations are perfect. This assumption might be unrealistic in some applications. To address this issue, we propose a new partial-observability model for online learning problems where the learner, besides its own loss, also observes some *noisy* feedback about the other actions, depending on the underlying structure of the problem. This problem might be seen as an instance of the framework in Figure 3.1. We represent the structure of the problem by a weighted directed graph, where the edge weights are related to the quality of the feedback shared by the connected nodes. For this problem, we propose two algorithms. In the first algorithm the learner simply specifies a threshold for "unreliable" side observations, discards them, and uses the rest of the

observations. However, selecting this threshold proves to be challenging. Therefore, in the second algorithm, we use a different approach. Instead of telling the algorithm which side observations are reliable and which are not, we use a novel approach to loss estimation which can control the bias of estimates of unreliable side observations.

For both algorithms, we guarantee a regret of $\widetilde{\mathcal{O}}(\sqrt{\alpha^* T})$ after T rounds, where α^* is a novel graph property that we call the *effective independence number*. To achieve this bound, the first algorithm needs to know the optimal threshold while the second algorithm is completely parameter-free and achieves this bound without any additional requirements.

For the special case of binary edge weights, our setting reduces to the partial-observability models of Mannor and Shamir [2011], studied in Section 2, and our algorithm recovers the near-optimal regret bounds.

As an illustration to our setting, consider the problem of controlling solar panels so as to maximize their power production. In this problem, the learner has to repeatedly decide about the orientation of the panels so as to find alignments with strong sunshine. Besides the amount of the energy being actually produced in the current alignment, the learner can also estimate the amount of energy in some other positions, based on measurements of sensors installed on the solar panel. However, the observations generated by these sensors can be of variable quality depending on visibility conditions, the quality of the sensors and the alignment of the panels. Overall, this problem can be seen as a bandit problem with noisy side observations fitting into our setting, where actions correspond to alignments and the noisy side observations give information about similar alignments.

Intuitively, in the case when the noise level of side observations does not change with time, a possible strategy one can think of is to use only the observations from the "most reliable" sources and ignore the rest. Having made the distinction between "reliable" and "unreliable", the learner could model the observation structure in the framework of Mannor and Shamir [2011], Alon et al. [2013], by treating every "reliable" observation as *perfect*. This approach raises two concerns. First, determining the cutoff for unreliable observations that allows the "most efficient" use of information is a highly nontrivial design choice. As we show later, knowing the *perfect cutoff* would help us to improve performance over the pure bandit setting without side observations. Second, one has to address the *bias* arising from handling every reliable observation as perfect. While one can think of many obvious ways to handle this bias by appropriate weighting observations, none of these solutions are directly compatible with the model of Mannor and Shamir [2011], Alon et al. [2013]. Our

main contribution in this section is an algorithm that is able to deal with both issues *without the knowledge of the optimal cutoff.*

The main tool we use for modeling uncertain observations is a *weighted directed graph* encoding the quality of side observations. In this graph, the weight of the arc $i \to j$ measures the quality of the side observation obtained from action j when selecting action i. All weights are assumed to lie in the interval $[0, 1]$, with a weight of 1 corresponding to a perfectly accurate side observation, and a weight of 0 corresponding to a side observation of useless noise. Our model generalizes the previously considered models of Mannor and Shamir [2011] and Alon et al. [2013]: their respective settings are captured by considering undirected and directed graphs with binary weights in our setting. In these special cases, the *independence number* α of the observation graph plays a key role in characterizing the complexity of learning: the minimax regret after T rounds is known to be $\widetilde{\Theta}(\sqrt{\alpha T})$. In this section, we define a similar quantity for weighted graphs: the *effective independence number* α^* and propose a learning algorithm that enjoys a regret bound of $\widetilde{\mathcal{O}}(\sqrt{\alpha^* T})$ without any conditions made on the loss sequence.

The effective independence number α^* is closely related to the cutoff threshold for noisy observations. Intuitively, it is linked to the independence number of a graph that only considers reliable observations. In practical scenarios, neither the cutoff nor α^* is ever known to the learner, which is the *main challenge* we need to address. In any case, the most interesting situations for our setting are the cases when we can bound α^* by a small quantity.

While we are mainly inspired by situations where the weights of the graph are fixed and known in advance, we treat a more general setting where the observation structure can arbitrarily change over time and the weights are revealed to the learner only after it has made its decision. Our algorithms are fully adaptive in the sense that they do not require any prior knowledge of the sequence of observation graphs or the time horizon. To achieve this result, we combine the *implicit exploration* strategy introduced in Section 1.3 with thresholding (in the first algorithm) or a loss estimation technique that effectively suppresses the observation noise (in the second algorithm).

For the special case of binary weights, the effective independence number and the independence number coincide; otherwise α^* is bounded by the number of actions N. Thus, the regret bound of our algorithm is of near-optimal order for binary graphs and is always within logarithmic factors of the minimax regret of order $\sqrt{NT}$ for the standard multi-armed bandit problem without side observations. As we will show later in the thesis, there are several interesting cases for which the effective

independence number can be bounded in a nontrivial way.

Independently of the work presented in this section, Wu, György, and Szepesvári [2015] considered an essentially identical partial-observability model for online learning: there, side observations are modeled as zero-mean Gaussian random variables with *variance* depending on the chosen action. It is easy to see that their model and ours can capture exactly the same type of problems: a side observation with zero variance in their model corresponds to a perfect observation with weight one in our model while useless noise is equivalently represented by infinite-variance or zero-weight observations. The results of Wu et al. [2015] are, however, of a completely different flavor than the ones presented in this work; the primary difference being that Wu et al. assume that the losses are i.i.d. Gaussian random variables while our results hold without any assumptions made on the sequence of losses. The main contributions of Wu et al. are (i) a general problem-dependent lower bound on the regret and (ii) algorithms that work under the assumption that all the useful (i.e., finite-variance) side observations have the same variance. This latter assumption does not use the full strength of the framework where the variance of side observations can vary for different actions. Notably, the regret bounds presented in this section match (up to logarithmic factors) the lower bounds of Wu et al. [2015] for the special cases that they consider. That said, their lower bounds and our upper bounds are not directly comparable for more general observability graphs.

Besides the works mentioned above, several other partial-observability models have been considered in the literature. The most general of these settings is the *partial-monitoring* framework considered by Bartók et al. [2011, 2014]. Unlike our model, this framework is most useful for identifying and handling feedback structures that are *more restrictive* than bandit feedback. In contrast, our framework deals with feedback structures that are strictly more expressive than plain bandit feedback. Similarly to Bartók et al., the recent work of Alon et al. [2015] also considers a generalization of the partial-observability models of Mannor and Shamir [2011] and Alon et al. [2013] that may be more restrictive than bandit feedback. Another well-studied setting in machine learning is where the observations are corrupted by noise irrespective of the decisions of the learner (see, e.g., Cesa-Bianchi et al., 2010). Such settings do not pose an exploration-exploitation dilemma to the learner and thus are not relevant to our goals.[1]

[1] In fact, it can be shown by the techniques of Devroye et al. [2013] that in the setting of online learning with finite actions and observations corrupted by the same level of i.i.d. noise, the simplest possible strategy of *following the leader* gives near-optimal guarantees.

4.1 Side-observation setting with weighted graphs

Let us now give the formal definition of our learning problem. We consider a sequential decision-making problem, which falls into the framework described in Figure 3.1, where a *learner* and an *environment* interact in the following way (see also Figure 3.7). In every round $t \in [T] = \{1, 2, \dots, T\}$, the environment selects a weighted graph G_t with N nodes and a loss function $\ell_t : [N] \to [0, 1]$ where $\ell_{t,i}$ is the loss associated with arm i. The weight of each arc $i \to j$ in G_t is denoted as $s_{t,(i,j)}$ and assumed to lie in $[0, 1]$. Following the environment's move, the learner selects an *action* (or *arm*) $I_t \in [N]$ and incurs the loss ℓ_{t,I_t}. Finally, the learner also observes G_t and the feedback

$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

for every arm i, where $\xi_{t,i}$ is the *observation noise* (c.f. another illustration on Figure 3.6). We assume that each $\xi_{t,i}$ is zero-mean, satisfies $|\xi_{t,i}| \leq R$ for some known constant $R \geq 0$, and is generated independently of all other noise terms and the history of the process[2]. The interaction history between the learner and the environment up to the end of round t is captured by the sigma-algebra $\mathcal{F}_t$. We consider *adaptive* (or *non-oblivious*) environments that are allowed to choose ℓ_t and G_t in full knowledge of the history $\mathcal{F}_{t-1}$. We also assume that all graphs G_t are such that $s_{t,(i,i)} = 1$ for all i, that is, the learner always observes its own loss ℓ_{t,I_t} without corruption.

The goal of the learner is to choose its actions so as to ensure that its cumulative loss grows as slowly as possible. As traditional in the online learning literature [Cesa-Bianchi and Lugosi, 2006], we measure the performance of the learner in terms of the (total expected) *regret* defined as the gap between the expected loss of the player and the expected loss of the best fixed-arm policy:

$$R_T = \max_{i \in [N]} \mathbb{E}\left[\sum_{t=1}^{T} \ell_{t,I_t} - \sum_{t=1}^{T} \ell_{t,i}\right].$$

We are interested in constructing algorithms for the learner that guarantees a tight upper bound on the regret. Before proposing our algorithms, a few comments are in order. First, notice that our framework technically contains the settings of Mannor

[2]We are mainly interested in the setting where $R = \Theta(1)$, that is, we are neither in the easy case where R is close to zero or the hard one where it may be as large as $\Omega(\sqrt{T})$

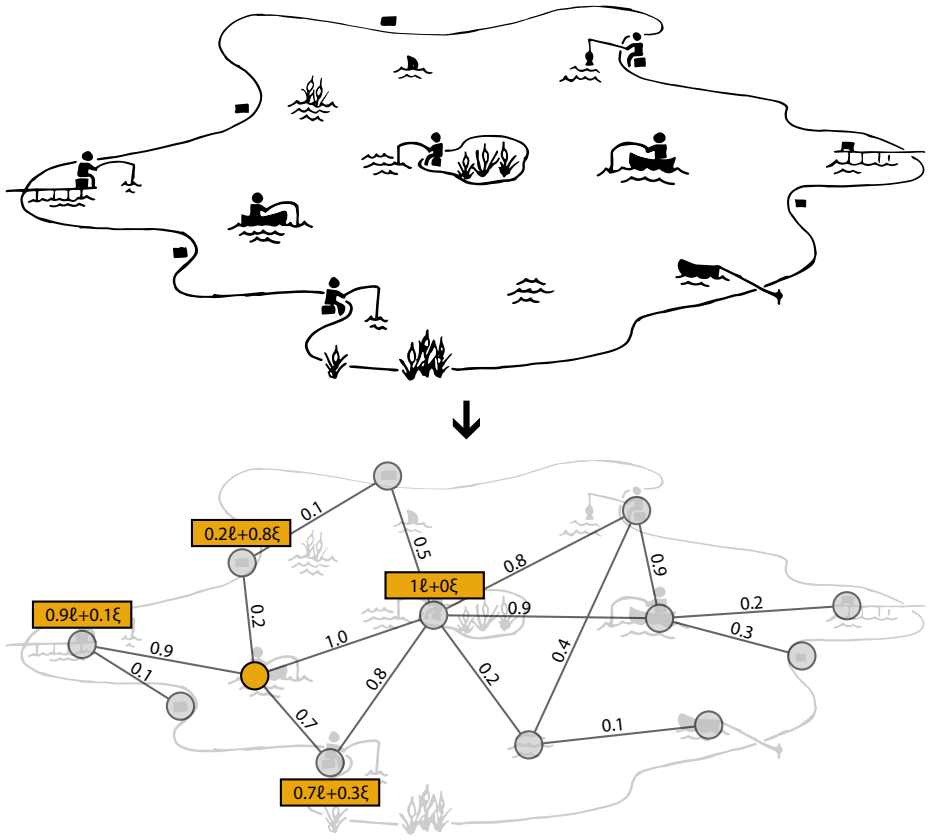


Figure 3.6: Noisy feedback on fishing example [Wu et al., 2015]: A fisherman picks a fishing spot daily and gets the yield while imperfectly observing the yields of neighbors.

and Shamir [2011] and Alon et al. [2013] as special cases where the edge weights are chosen from $\{0, 1\}$: in this situation, our framework suggests that the learner either gets *perfect* side-observations or just zero-mean noise, which can be safely ignored by the learner. Also, notice that since we assume $s_{t,(i,i)} = 1$ for all i, our problem is not harder for the learner than the standard multi-armed bandit problem. Indeed, thanks to this property, the learner could simply ignore all side observations and run a bandit algorithm such as Exp3 of Auer et al. [2002a] that guarantees a regret bound of $\widetilde{\mathcal{O}}(\sqrt{NT})$.

In what follows, we present learning algorithms with strong theoretical performance guarantees for the setting described in the previous section and a new quantity, effective independence number, that characterizes the connectivity of a graph. As the intuitions underlying our algorithm are rather intricate, we will proceed gradually: we

Parameters:
  Known set of actions $[N]$
  Not necessarily known time horizon T.
for $t = 1$ **to** T **do**
  The environment chooses a loss function $\ell_t : [N] \to [0, 1]$
  The adversary chooses a directed weighted graph G_t with edge weights in $[0, 1]$.
  Based on its previous observations (and possibly some source of randomness), the learner picks an action $I_t \in [N]$.
  The learner suffers loss ℓ_{t,I_t}.
  The learner observes G_t and the feedback

$$c_{t,i} = s_{t,(I_t,i)} \cdot \ell_{t,i} + \left(1 - s_{t,(I_t,i)}\right) \cdot \xi_{t,i}$$

  for every arm $i \in [N]$, where $\xi_{t,i} \in [-R, R]$ is the zero-mean independent observation noise.
end for

Figure 3.7: The protocol of online learning with noisy observations.

first identify the main challenges of constructing learning algorithms for our setting, then offer a solution that overcomes these difficulties in an efficient manner.

4.2 EXP3-IXT algorithm and theoretical guarantees

We first consider an algorithm that bases its decisions on the following estimates of each $\ell_{t,i}$:

$$\hat{\ell}_{t,i}^{(\mathrm{B})} = \frac{c_{t,i}}{\sum_{j \in N_i^-} p_{t,j} s_{t,(j,i)} + \gamma_t}, \tag{3.10}$$

where B stands for "basic". Here, $\gamma_t \geq 0$ is an *implicit exploration* parameter [Kocák et al., 2014a], introduced in Section 1.3, for decreasing the variance of importance-weighted estimates. Notice that setting $\gamma_t = 0$, makes estimates above unbiased since

$$\mathbb{E}\left[c_{t,i} \middle| \mathcal{F}_{t-1}\right] = \left(\sum_{j=1}^{N} p_{t,j} s_{t,(j,i)}\right) \cdot \ell_{t,i},$$

where we used our assumption that $\mathbb{E}[\xi_{t,i}] = 0$. Using these estimates in our EXP3 algorithmic template (Algorithm 4), one would expect to get reasonable performance guarantees. Unfortunately, we were not able to prove a performance guarantee for the resulting algorithm.

A close examination reveals that the reason for the poor performance of the above algorithm is the large variance of the estimates (3.10) which is caused by including observations from "unreliable sources" with small weights. One intuitive idea is to explicitly draw the line between reliable and unreliable sources by cutting connections with weights under a certain threshold. This effect is realized by the estimates

$$\hat{\ell}_{t,i}^{(\mathrm{T})} = \frac{c_{t,i}\mathbb{1}\left\{s_{t,(I_t,i)} \geq \varepsilon_t\right\}}{\sum_{j\in N_i^-} p_{t,j}s_{t,(j,i)}\mathbb{1}\left\{s_{t,(j,i)} \geq \varepsilon_t\right\} + \gamma_t}, \tag{3.11}$$

where $\varepsilon_t \in [0,1]$ is a threshold value and T stands for "thresholded". We call the algorithm resulting from using the above estimates in Algorithm 4 EXP3-IXT, standing for "EXP3 with Implicit eXploration and Truncated side-observation weights". Thanks to the thresholding operation, the variance of the loss estimates can be nicely controlled and it becomes possible to prove a strong performance guarantee for EXP3-IXT. In particular, using techniques similar to the analysis in Section 1.4, we prove the following result concerning the regret of EXP3-IXT algorithm:

Theorem 9. *For all $t \in [T]$, let ε_t be a threshold used by* EXP3-IXT *algorithm at time t. Setting*

$$\eta_t = \sqrt{\frac{\log N}{2(1+R^2)\left(\frac{N}{\varepsilon_t} + \sum_{s=1}^{t-1}\frac{Q_s^{\mathrm{IXT}}}{\varepsilon_s}\right)}} \qquad \text{and} \qquad \gamma_t = \frac{1+R^2}{2\varepsilon_t}\eta_t$$

the cumulative regret of EXP3-IXT *is bounded as*

$$R_t \leq \mathbb{E}\left[\sqrt{8(1+R^2)(\log N)\left(N + \sum_{t=1}^{T}\frac{Q_t^{\mathrm{IXT}}}{\varepsilon_t}\right)}\right],$$

where $Q_t^{\mathrm{IXT}} = Q_t(1,0,\gamma_t)$ for graph G_t thresholded by ε_t

The theorem is proved later in Section 6.3. To obtain a deterministic bound, we can

Algorithm 7 EXP3-IXT

Input and initialization:
  Set of actions $\mathcal{A} = [N]$, time horizon T
  Initialize cumulative loss estimates $\widehat{L}_{0,i}$ to 0 for all $i \in [N]$
for $t = 1$ **to** T **do**
  The adversary privately chooses losses $\ell_{t,i}$ for $i \in [N]$ and generates graph G_t
  Set threshold $\varepsilon_t \in [0, 1]$, possibly using G_t
  Set implicit exploration term γ_t and adaptive learning rate η_t as

$$\eta_t = \sqrt{\frac{\log N}{2(1+R^2)\left(\frac{N}{\varepsilon_t} + \sum_{s=1}^{t-1} \frac{Q_s^{\mathrm{IX_T}}}{\varepsilon_s}\right)}} \qquad \text{and} \qquad \gamma_t = \frac{1+R^2}{2\varepsilon_t}\eta_t$$

  Create exponential weights $w_{t,i} = \frac{1}{N}\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for all $i \in [N]$
  Create probability distribution $p_{t,i} = \frac{w_{t,i}}{W_t}$ where $W_t = \sum_{i=1}^{N} w_{t,i}$
  Choose an action I_t such that $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,N})$
  Incur and observe the loss of the action I_t
  Observe noisy side observations $c_{t,i} = s_{I_t,i}\ell_{t,i} + (1 - s_{I_t,i})\xi_{t,i}$ for all $i \in [N]$
  Construct loss estimates for every action $i \in [N]$, such that

$$\hat{\ell}_{t,i} = \frac{c_{t,i}}{o_{t,i} + \gamma_t}\mathbb{1}_{\{(I_t \to i) \in G_t\}} \qquad \text{where} \qquad o_{t,i} = \sum_{j \in [N]} p_{t,j} s_{j,i}$$

end for

use Lemma 22 to upper bound $Q_t^{\mathrm{IX_T}} = Q_t(1, 0, \gamma_t)$ as

$$Q_t^{\mathrm{IX_T}} \leq 2\frac{\alpha_t(\varepsilon_t)}{\varepsilon_t}\left(1 + \log\left(1 + \frac{N^2 + 2N\gamma_t}{\gamma_t \alpha_t(\varepsilon_t)}\right)\right). \tag{3.12}$$

Using this bound we obtain the main result for EXP3-IXT algorithm in the form of the following corollary

Corollary 3. *The regret of* EXP3-IXT *satisfies*

$$R_t \leq \sqrt{8(1+R^2)(\log N)\left(N + \sum_{t=1}^{T} H_t \frac{\alpha_t(\varepsilon_t)}{\varepsilon_t^2}\right)}$$

where

$$H_t = 2 + 2\log\left(1 + \frac{N^2\varepsilon_t\sqrt{\frac{8(1+R^2)}{\log N}\left(\sum_{s=1}^{t}\frac{N}{\varepsilon_t}\right)} + 2N}{\alpha_t(\varepsilon_t)}\right)$$

Proof. The proof is obtained using bound (3.12) together with the definition of γ_t. Moreover we bound every Q_t^{IXT} in the definition of γ_t by N. □

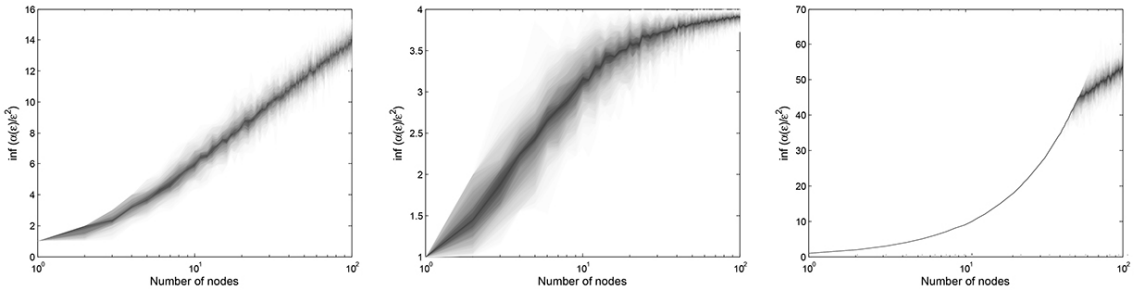
Note that the regret bound of EXP3-IXT is of $\widetilde{\mathcal{O}}\left(\sqrt{\sum_{t=1}^{T}\frac{\alpha_t(\varepsilon_t)}{\varepsilon_t^2}}\right)$. In order to optimize this bound we can choose $\varepsilon_t = \arg\min_{\varepsilon\in[0,1]}\frac{\alpha_t(\varepsilon)}{\varepsilon^2}$. We denote this optimal value of ε_t by ε_t^*. Note however that finding ε_t^* can be a very challenging task in practice since computing independence number, in general, is known to be NP-hard. Even worse, computing ε_t^* for a weighted graph can require computing up to N^2 independence numbers. In the next section, we discuss about optimal threshold and define a quantity which characterizes the complexity of the problem. Later in this chapter, we show a computationally more efficient algorithm for the setting which does not need to know the optimal threshold ε_t^*.

4.3 Effective independence number

In the previous section, we showed that the regret bound for EXP3-IXT algorithm is of order $\widetilde{\mathcal{O}}\Big(\sqrt{\sum_{t=1}^{T}\frac{\alpha_t(\varepsilon_t)}{\varepsilon_t^2}}\Big)$. Optimizing this bound and choosing $\varepsilon_t = \varepsilon_t^* = \arg\min_{\varepsilon\in[0,1]}\frac{\alpha_t(\varepsilon)}{\varepsilon^2}$ motivates us to define a new graph property that we call *effective independence number*, defined as follows:

Definition 11. *Let G be a weighted directed graph with N nodes and edge weights bounded in $[0,1]$. For all $\varepsilon\in[0,1]$, let $G(\varepsilon)$ be the (unweighted) directed graph where arc $i\to j$ is present if and only if $s_{i,j}\geq\varepsilon$ in G. Letting $\alpha(\varepsilon)$ be the independence number of $G(\varepsilon)$, the effective independence number of G is defined as*

$$\alpha^* = \min_{\varepsilon\in[0,1]}\frac{\alpha(\varepsilon)}{\varepsilon^2}.$$



(a) $U(0,1)$ weights (b) $U(\frac{1}{2},1)$ weights (c) $U(0,\frac{1}{2})$ weights

Figure 3.8: Dependence of α^* on the size of the graph with random weights, 100 graphs for each size.

Roughly speaking, the effective independence number is a measure of connectivity of weighted graphs. Using the notion of the effective independence number, we can show that, using optimal thresholds $\varepsilon_t = \varepsilon_t^*$ in every round $t \in [T]$, Exp3-IXt algorithm enjoys the regret bound of $\widetilde{\mathcal{O}}\Big(\sqrt{\sum_{t=1}^{T} \alpha_t^*}\Big)$.

The previous section has established that the performance guarantees of our algorithms can be expressed in terms of the effective independence number of the observation graphs. In this section, we provide some basic insights about the nature of this quantity and describe some graph structures with small effective independence numbers.

The first observation we make is that the effective independence number is always well-defined, as the function $\alpha(\varepsilon)/\varepsilon^2$ can be easily shown to be piecewise decreasing and lower semicontinuous with at most N discontinuities. Thanks to these properties, this expression takes its minimum within the closed interval $[0,1]$. Second, we note that the effective independence number of any weighted graph is trivially bounded by the number N of the nodes in the graph. This follows from the fact that $\alpha^* \leq \alpha(1)/1 \leq N$. This essentially guarantees that incorporating side-observations can never be harmful to the performance of the learner: the regret of Exp3-WIX is always within logarithmic factors of the minimax regret of order $\sqrt{NT}$ for the standard multi-armed bandit problem without side observations.

It is also easy to see that the effective independence number exactly matches the independence number if all edge weights are binary. This, in particular, implies that for such graphs, the regret of Exp3-WIX grows at the minimax rate established by Alon et al. [2013] up to logarithmic factors, matching the performance guarantees of the algorithms of Alon et al. [2013] and Kocák et al. [2014a]. Another interesting

case is when all weights are either zero or equal to a fixed constant ε, also assuming $s_{i,i} = \varepsilon$. In this case, the effective independence number becomes $\frac{\alpha}{\varepsilon^2}$, where α is the independence number of the underlying unweighted graph. This case was studied in the recent paper of Wu et al. [2015], who show (in their Corollary 4) that the *minimax* regret in this case is of $\Theta(\sqrt{\alpha T}/\varepsilon)$—implying that our performance bounds for this case are again near-optimal[3]. Also, observe that whenever all weights are bounded by some constant $c > 0$ from below, the effective independence number becomes upper-bounded by $1/c^2$, *irrespective of the number of actions*. That is, our algorithm can achieve an *exponential* performance gain over bandit algorithms in terms of N by leveraging such feedback structures.

Let us now describe a class of weighted graphs with bounded effective independence numbers. Consider a geometric graph whose nodes represent vertices of a uniform $k \times k$ grid on $[0,1]^2$. The weight of edge (i,j) is given as $1/(1+d_{i,j}^2)$, where $d_{i,j}$ is the Euclidean distance of the respective vertices represented by i and j. This graph can be used to model a sensor network where the measurement accuracy of measurements degrades with the distance. Thus, reading the measurements from one sensor will give information about the measurements of nearby sensors as well. Intuitively, increasing the number of sensors (i.e., refining the grid) should only improve the information-sharing between sensors up to a certain level. It is natural to expect a reasonable graph property quantifying the information-sharing efficiency to capture this intuition. We have numerically evaluated the effective independence number of a number of graphs from the above family to test if it satisfies the above criterion. We have found that the effective independence numbers remain bounded by a *constant* (roughly 30) even when refining the grid infinitely, confirming that the effective independence number captures the above phenomenon.

Finally, we conducted some numerical simulations to evaluate the average effective independence numbers of certain types of weighted random graphs. In particular, we considered random graphs with i.i.d. weights distributed uniformly on $[0,1]$, $[\frac{1}{2},1]$ and $[0,\frac{1}{2}]$. The distributions of the effective independence numbers are illustrated as scatter plots for different graph sizes on Figure 3.8. First, observe that the average α^* of $U(0,1)$-weighted graphs shows a logarithmic trend in terms of N. The results concerning $U(\frac{1}{2},1)$-weighted graphs are not surprising given that we have already established that graphs with bounded weights have finite effective independence numbers. For $U(0,\frac{1}{2})$-weighted graphs, we see that α^* grows linearly up until a certain threshold when it starts to follow a logarithmic trend. The intuition behind

[3]While we prove our bounds for the case where $s_{i,i} = 1$ for all i, it is easy to extend our results to the case where all such weights equal a constant in $[0,1]$.

this linear behavior for small graphs is the following. First, observe that the optimal value of ε is greater than $1/\sqrt{N}$. That is, until N is large enough so that a critical mass of edges are above this quantity, the optimal value of $\alpha(\varepsilon)/\varepsilon^2$ remains N. Once N is beyond this critical value, α^* starts following a logarithmic trend.

4.4 EXP3-WIX algorithm and theoretical guarantees

One downside of the EXP3-IXT algorithm is the necessity of choosing a threshold ε_t. In this section, we present our main algorithm for the setting with noisy side observations that obtains strong regret bound, similar to the bound of EXP3-IXT, without having to compute the best threshold ε_t^*, nor effective independence numbers. The key element of this algorithm is using specifically designed loss estimates of the form

$$\hat{\ell}_{t,i} = \frac{s_{t,(I_t,i)} \cdot c_{t,i}}{\sum_{j \in N_i^-} p_{t,j} s_{t,(j,i)}^2 + \gamma_t}, \tag{3.13}$$

where $\gamma_t \geq 0$ is the implicit exploration parameter already introduced in Section 1.3. Notice that the difference from the estimates (3.10) is that the observation $c_{t,i}$ is multiplied by the weight of useful information in $c_{t,i}$ and the denominator is modified accordingly, so that the estimates are unbiased when setting $\gamma_t = 0$ since

$$\mathbb{E}\left[s_{t,(I_t,i)} \cdot c_{t,i} \middle| \mathcal{F}_{t-1} \right] = \left(\sum_{j=1}^{N} p_{t,j} s_{t,(j,i)}^2 \right) \cdot \ell_{t,i}.$$

The role of this scaling is pulling the noise term $\xi_{t,i}$ toward zero for actions i with small weights $s_{I_t,i}$, and thus achieving a similar variance-reducing effect as the truncations employed by EXP3-IXT.

Armed with the loss estimates (3.13), we are ready to define our algorithm: EXP3 (presented as Algorithm 4) with Weighted observations and Implicit eXploration, or, in short, EXP3-WIX. Overall, EXP3-WIX has two set of parameters to tune: the sequence of learning rates $(\eta_t)_t$ and the sequence of IX parameters $(\gamma_t)_t$. Our main theorem below states the performance guarantees of EXP3-WIX with an adaptive learning-rate sequence that does not need any prior knowledge about the number of rounds or the nature of the side-observation graphs. The key quantity for computing

Algorithm 8 EXP3-WIX

Input and initialization:
  Set of actions $\mathcal{A} = [N]$, time horizon T
  Initialize cumulative loss estimates $\widehat{L}_{0,i}$ to 0 for all $i \in [N]$
for $t = 1$ **to** T **do**
  The adversary privately chooses losses $\ell_{t,i}$ for $i \in [N]$ and generates graph G_t
  Set implicit exploration term γ_t and adaptive learning rate η_t as

$$\eta_t = \sqrt{\frac{\log N}{2(1+R^2)\left(N + \sum_{s=1}^{t-1} Q_s^{\mathrm{WIX}}\right)}} \qquad \text{and} \qquad \gamma_t = \frac{1+R^2}{2}\eta_t$$

  Create exponential weights $w_{t,i} = \frac{1}{N}\exp\left(-\eta_t \widehat{L}_{t-1,i}\right)$ for all $i \in [N]$
  Create probability distribution $p_{t,i} = \frac{w_{t,i}}{W_t}$ where $W_t = \sum_{i=1}^{N} w_{t,i}$
  Choose an action I_t such that $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,N})$
  Incur and observe the loss of the action I_t
  Observe noisy side observations $c_{t,i} = s_{I_t,i}\ell_{t,i} + (1 - s_{I_t,i})\xi_{t,i}$ for all $i \in [N]$
  Using G_t construct loss estimate for every action $i \in [N]$, such that

$$\hat{\ell}_{t,i} = \frac{s_{I_t,i} c_{t,i}}{o_{t,i} + \gamma_t} \mathbb{1}_{\{(I_t \to i) \in G_t\}} \qquad \text{where} \qquad o_{t,i} = \sum_{j \in [N]} p_{t,j} s_{j,i}^2$$

end for

the parameters η_t and γ_t is

$$Q_t^{\mathrm{WIX}} = \sum_{i=1}^{N} \frac{p_{t,i}}{\sum_{j=1}^{N} p_{t,j} s_{t,(j,i)}^2 + \gamma_t},$$

defined for all t. Notice that Q_t^{WIX} is defined as $Q(1, 1, \gamma_t)$ for graph G_t. We can use the definition of Q_t^{WIX} to characterize the performance guarantees of EXP3-WIX algorithm in the following theorem.

Theorem 10. *Setting*

$$\eta_t = \sqrt{\frac{\log N}{2(1+R^2)\left(N + \sum_{s=1}^{t-1} Q_s^{\mathrm{WIX}}\right)}} \qquad \textit{and} \qquad \gamma_t = \frac{1+R^2}{2}\eta_t$$

for all $t \in [T]$*, the cumulative regret* R_T *of* Exp3-WIX *algorithm is bounded as*

$$R_T \leq \mathbb{E}\left[\sqrt{8(1+R^2)(\log N)\left(N + \sum_{t=1}^{T} Q_t^{\mathrm{WIX}}\right)}\right].$$

The proof of the theorem is located in the Section 6.4.

The next step is to find a deterministic upper bound on Q_t^{WIX}. For this purpose, we use the fact that $Q_t^{\mathrm{WIX}} = Q_t(1, 1, \gamma_t)$ and bound this quantity using Lemma 22 to obtain

$$Q_t^{\mathrm{WIX}} \leq 2\alpha_t^* \left[1 + \log\left(1 + \frac{2N}{(\varepsilon_t^*)^2 \alpha_t^*} + \frac{N^2}{\gamma_t \alpha_t^*}\right)\right], \tag{3.14}$$

where ε_t^* is an optimal threshold. This gives us following corollary which characterizes the regret of Exp3-WIX algorithm in the terms of effective independence number.

Corollary 4. *The regret of* Exp3-WIX *satisfies*

$$R_t \leq \sqrt{8(1+R^2)(\log N)\left(N + 2\sum_{t=1}^{T} H_t \alpha_t^*\right)}$$

where

$$H_t = \alpha_t^* \left[1 + \log\left(1 + \frac{2N}{(\varepsilon_t^*)^2 \alpha_t^*} + \frac{N^2}{\alpha_t^*}\sqrt{\frac{8Nt}{(1+R^2)\log N}}\right)\right].$$

Proof. To proof this corollary, we use Theorem 10, bound (3.14) on Q_t^{WIX}, and the definition of γ_t in which we bound every appearance of Q_t^{WIX} by N. □

In plain words, Corollary 4 guarantees that the regret of Exp3-WIX grows as $\widetilde{\mathcal{O}}\left(\sqrt{\sum_{t=1}^{T} \alpha_t^*}\right) = \widetilde{\mathcal{O}}(\sqrt{\alpha_{\mathrm{avg}}^* T})$. Notice that in order to obtain this regret bound, Exp3-WIX never needs to compute the effective independence number of any of the observation graphs. This saves us from a significant computational overhead as compared to the naïve algorithm Exp3-IXt that needed to set a thresholding parameter to discard unreliable observations.

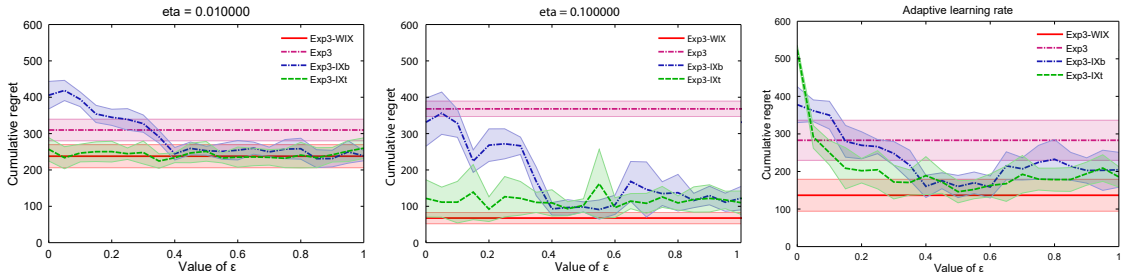


Figure 3.9: Comparison of total regrets of the algorithms at time T for static and adaptive learning rates.

4.5 Experiments

In this section, we empirically compare Exp3-WIX to some of its natural competitors: Exp3-IXt, vanilla Exp3 that ignores all side observations and a straightforward variation of the Exp3-IX algorithm. This latter algorithm, referred to as Exp3-IXb (with "b" standing for "basic"), uses a threshold ε_t to decide which observations are too noisy to use and which are the ones to be retained: All the edges with weights smaller than a parameter ε are deleted and the rest of the weights are set to 1. The algorithm then plays basic Exp3-IX for the resulting binary graph. That is, the difference between Exp3-IXt and Exp3-IXb is that the latter does not adjust for the bias arising from using unreliable side observations. Note that Exp3-IXb comes without any formal performance guarantee.

For the purpose of the experiments, we assumed to have 25 actions forming 5×5 grid embedded in a plane. The distance of neighbors in the grid was set to be 1. Using this structure, we defined the weight connecting two nodes as $\min\{3/d^2, 1\}$, and d is the Euclidean distance between actions in the grid.

For constructing the loss sequence, we interleaved 20 Gaussian random walks with small increments for each action, with appropriate truncations to keep the losses within the $[0, 1]$ interval. Using this procedure, we generated a single loss sequence of $T = 5,000$ steps to test the algorithms. For a fair comparison, we ran each algorithm for their respective theoretically motivated adaptive learning rates, and also for a number of static learning rates. For static learning rates, we observed the best performance of Exp3 for learning rates around 0.01, all the other algorithms did well for learning rates around 0.1.

We ran Exp3-IXb and Exp3-IXt for several values of ε from 0 to 1. In all ex-

periments, we set the implicit exploration parameters to zero. This is well-justified in the case of undirected graphs, as shown by the analysis of Alon et al. [2013]. Figure 3.9 shows the performance of the algorithms as a function of the threshold parameter ε. Each curve on this graph is the average of the total regrets measured in 10 independent runs with error bars proportional to the empirical standard deviation.

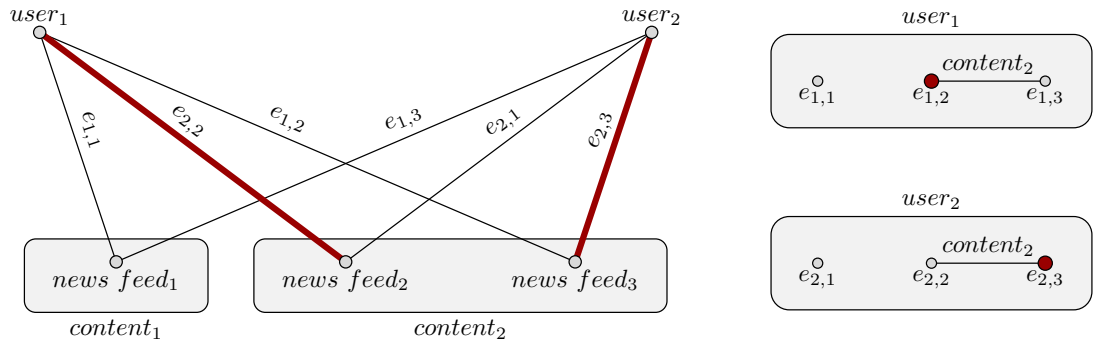
Our experiments confirm that guessing the right value for the threshold parameter is indeed a very difficult problem: while Exp3-WIX performs consistently well for all parameter settings, Exp3-IXt and Exp3-IXb only perform reasonably well for moderate values of ε that are not supported by theory. (In fact, the graph is designed so that the value of ε optimizing $\alpha(\varepsilon)/\varepsilon^2$ is 1, which suggests that only observations from immediate neighbors in the grid are relevant.) Perhaps surprisingly, Exp3-IXb performs well despite the obvious bias in its loss estimates. The performance of Exp3 is significantly worse than Exp3-WIX, confirming the benefit of side-observations, however noisy they are.

5 Combinatorial semi-bandits with adversarial side observations

In a multi-armed bandit problem, the learner plays one node at the time. This might be too restrictive for some applications. Therefore, in this section, we generalize the partial observability model introduced by Mannor and Shamir [2011] (Figure 3.1) to the combinatorial case where the learner can play several nodes at the same time. Similarly to the previous sections, we use observability graphs to capture the complexity of the feedback. From a different point of view, we can see this problem as an interpolation between semi-bandit problem, where the learner observes only losses of the selected nodes, and full-information setting where the learner observes all the losses regardless of his action.

5.1 Introduction

Consider the problem of sequentially recommending content for a set of users. In each period of this online decision problem, we have to assign content from a news feed to each of our subscribers so as to maximize clickthrough. We assume that this assignment needs to be done well in advance so that we only observe the actual



(a) Users and news feeds. The thick edges represent one potential matching of users to feeds, grouped news feeds show the same content.

(b) Users and news feeds. Connected feeds mutually reveal each others click-throughs.

Figure 3.10: Combinatorial bandits example

content after the assignment was made and the user had the opportunity to click. While we can easily formalize the above problem in the classical multi-armed bandit framework [Auer et al., 2002b], notice that we will be throwing out important information if we do so! The additional information in this problem comes from the fact that several news feeds can refer to the same content, giving us the opportunity to infer clickthroughs for a number of assignments that we *did not actually make*. For example, consider the situation shown on Figure 3.10a. In this simple example, we want to suggest one out of three news feeds to each user, that is, we want to choose a matching on the graph shown on Figure 3.10b which covers the users. Assume that news feeds 2 and 3 refer to the same content, so *whenever we assign news feed 2 or 3 to any of the users, we learn the value of both of these assignments*. The relations between these assignments can be described by a graph structure (shown on Figure 3.10b), where nodes represent user-news feed assignments, and edges mean that the corresponding assignments reveal the clickthroughs of each other. For a more compact representation, we can group the nodes by the users, and rephrase our task as having to choose one node from each group. Besides its own reward, each selected node reveals the rewards assigned to all their neighbors.

The problem described above fits into the framework of *online combinatorial optimization* where in each round, a learner selects one of a very large number of available actions so as to minimize the losses associated with its sequence of decisions. Various instances of this problem have been widely studied in recent years under different feedback assumptions Cesa-Bianchi and Lugosi [2012], Audibert et al. [2014], Chen

et al. [2013], notably including the so-called *full-information* Koolen et al. [2010] and *semi-bandit* Audibert et al. [2014], Neu and Bartók [2013] settings. Using the example in Figure 1a, assuming full information means that clickthroughs are observable for *all* assignments, whereas assuming semi-bandit feedback, clickthroughs are only observable on the actually realized assignments. While it is unrealistic to assume full feedback in this setting, assuming semi-bandit feedback is far too restrictive in our example. Similar situations arise in other practical problems such as packet routing in computer networks where we may have additional information on the delays in the network besides the delays of our own packets.

5.2 Combinatorial side-observation setting with adversarial graphs

We now turn our attention to the setting of online combinatorial optimization (see Koolen et al. [2010], Cesa-Bianchi and Lugosi [2012], Audibert et al. [2014]). In this variant of the online learning problem, the learner has access to a possibly huge action set $\mathcal{S} \subseteq \{0,1\}^N$ where each action is represented by a binary vector $\boldsymbol{v}$ of dimensionality N. In what follows, we assume that $\|\boldsymbol{v}\|_1 \leq m$ holds for all $\boldsymbol{v} \in \mathcal{S}$ and some $1 \leq m \ll N$, with the case $m = 1$ corresponding to the multi-armed bandit setting considered in the previous section. In each round $t = 1, 2, \ldots, T$ of the decision process, the learner picks an action $\boldsymbol{V}_t \in \mathcal{S}$ and incurs a loss of $\boldsymbol{V}_t^\intercal \boldsymbol{\ell}_t$. At the end of the round, the learner receives some feedback based on its decision $\boldsymbol{V}_t$ and the loss vector $\boldsymbol{\ell}_t$. The regret of the learner is defined as

$$R_T = \max_{\boldsymbol{v} \in \mathcal{S}} \mathbb{E}\left[\sum_{t=1}^{T} (\boldsymbol{V}_t - \boldsymbol{v})^\intercal \boldsymbol{\ell}_t\right].$$

Previous work has considered the following feedback schemes in the combinatorial setting:

- The full-information scheme where the learner gets to observe $\boldsymbol{\ell}_t$ regardless of the chosen action. The minimax optimal regret of order $m\sqrt{T \log N}$ here is achieved by the ComponentHedge algorithm of Koolen et al. [2010], while the Follow-the-Perturbed-Leader (FPL) algorithm [Kalai and Vempala, 2005,

Hannan, 1957] was shown to enjoy a regret of order $m^{3/2}\sqrt{T \log N}$ by [Neu and Bartók, 2013].

- The semi-bandit scheme where the learner gets to observe the components $\ell_{t,i}$ of the loss vector where $V_{t,i} = 1$, that is, the losses along the components chosen by the learner at time t. As shown by Audibert et al. [2014], COMPONENT-HEDGE achieves a near-optimal $\mathcal{O}(\sqrt{mNT \log N})$ regret guarantee, while Neu and Bartók [2013] show that FPL enjoys a bound of $\mathcal{O}(m\sqrt{NT \log N})$.
- The bandit scheme where the learner only observes its own loss $\boldsymbol{V}_t^\top \boldsymbol{\ell}_t$. There are currently no known efficient algorithms that get close to the minimax regret in this setting—the reader is referred to Audibert et al. [2014] for an overview of recent results.

In this section, we define a new feedback scheme situated between the semi-bandit and the full-information schemes. In particular, we assume that the learner gets to observe the losses of some other components not included in its own decision vector $\boldsymbol{V}_t$. Similarly to the model of Alon et al. [2013], the relation between the chosen action and the side observations are given by a directed observability G_t (see example in Figure 1). We refer to this feedback scheme as *semi-bandit with side observations*. While our theoretical results stated in Section 2.3 continue to hold in this setting, combinatorial EXP3-IX could rarely be implemented efficiently—we refer to Cesa-Bianchi and Lugosi [2012], Koolen et al. [2010] for some positive examples. As one of the main concerns in this paper is computational efficiency, we take a different approach: we propose a variant of FPL that efficiently implements the idea of implicit exploration in combinatorial semi-bandit problems with side observations.

5.3 Implicit exploration by geometric resampling and FPL-IX algorithm

In each round t, FPL bases its decision on some estimate $\widehat{\boldsymbol{L}}_{t-1} = \sum_{s=1}^{t-1} \hat{\boldsymbol{\ell}}_s$ of the total losses $\boldsymbol{L}_{t-1} = \sum_{s=1}^{t-1} \boldsymbol{\ell}_s$ as follows:

$$\boldsymbol{V}_t = \operatorname*{arg\,min}_{\boldsymbol{v} \in \mathcal{S}} \boldsymbol{v}^\top \left(\eta_t \widehat{\boldsymbol{L}}_{t-1} - \boldsymbol{Z}_t \right). \tag{3.15}$$

Here, $\eta_t > 0$ is a parameter of the algorithm and $\boldsymbol{Z}_t$ is a perturbation vector with components drawn independently from an exponential distribution with unit expectation. The power of FPL lies in that it only requires an oracle that solves the

(offline) optimization problem $\min_{\boldsymbol{v}\in\mathcal{S}} \boldsymbol{v}^\mathsf{T}\boldsymbol{\ell}$ and thus can be used to turn any efficient offline solver into an online optimization algorithm with strong guarantees. To define our algorithm precisely, we need some further notation. We redefine $\mathcal{F}_{t-1}$ to be $\sigma(\boldsymbol{V}_{t-1},\dots,\boldsymbol{V}_1)$, $O_{t,i}$ to be the indicator of the observed *component* and let

$$q_{t,i} = \mathbb{E}\left[V_{t,i}\,|\mathcal{F}_{t-1}\right] \qquad \text{and} \qquad o_{t,i} = \mathbb{E}\left[O_{t,i}\,|\mathcal{F}_{t-1}\right].$$

The most crucial point of our algorithm is the construction of our loss estimates. To implement the idea of implicit exploration by optimistic biasing, we apply a modified version of the geometric resampling method of Neu and Bartók [2013] constructed as follows: Let $\boldsymbol{O}'_t(1), \boldsymbol{O}'_t(2), \dots$ be independent copies[4] of $\boldsymbol{O}_t$ and let $U_{t,i}$ be geometrically distributed random variables for all $i = [N]$ with parameter γ_t. We let

$$K_{t,i} = \min\left(\left\{k : O'_{t,i}(k) = 1\right\} \cup \{U_{t,i}\}\right) \tag{3.16}$$

and define our loss-estimate vector $\hat{\boldsymbol{\ell}}_t \in \mathbb{R}^N$ with its i-th element as

$$\hat{\ell}_{t,i} = K_{t,i} O_{t,i} \ell_{t,i}. \tag{3.17}$$

By definition, we have $\mathbb{E}\left[K_{t,i}\,|\mathcal{F}_{t-1}\right] = 1/(o_{t,i} + (1-o_{t,i})\gamma_t)$, implying that our loss estimates are *optimistic* in the sense that they lower bound the losses in expectation:

$$\mathbb{E}\left[\hat{\ell}_{t,i}\middle|\,\mathcal{F}_{t-1}\right] = \frac{o_{t,i}}{o_{t,i} + (1-o_{t,i})\gamma_t}\ell_{t,i} \le \ell_{t,i}.$$

Here we used the fact that $O_{t,i}$ is independent of $K_{t,i}$ and has expectation $o_{t,i}$ given $\mathcal{F}_{t-1}$. We call this algorithm Follow-the-Perturbed-Leader with Implicit eXploration (FPL-IX, Algorithm 9).

Note that the geometric resampling procedure can be terminated as soon as $K_{t,i}$ becomes well-defined for all i with $O_{t,i} = 1$. As noted by Neu and Bartók [2013], this requires generating at most N copies of $\boldsymbol{O}_t$ on expectation. As each of these copies requires one access to the linear optimization oracle over $\mathcal{S}$, we conclude that the expected running time of FPL-IX is at most N times that of the expected running time of the oracle. A high-probability guarantee of the running time can be obtained

[4]Such independent copies can be simply generated by sampling independent copies of $\boldsymbol{V}_t$ using the FPL rule (3.15) and then computing $\boldsymbol{O}'_t(k)$ using the observability G_t. Notice that this procedure requires no interaction between the learner and the environment, although each sample requires an oracle access.

by observing that $U_{t,i} \leq \log\left(\frac{1}{\delta}\right)/\gamma_t$ holds with probability at least $1-\delta$ and thus we can stop sampling after at most $N\log\left(\frac{N}{\delta}\right)/\gamma_t$ steps with probability at least $1-\delta$.

Algorithm 9 FPL-IX

Input: Set of actions $\mathcal{S}$,
  parameters $\gamma_t \in (0,1)$, $\eta_t > 0$ for $t \in [T]$.
for $t = 1$ **to** T **do**
  An adversary privately chooses losses $\ell_{t,i}$ for all $i \in [N]$ and generates a graph G_t
  Draw $Z_{t,i} \sim \mathrm{Exp}(1)$ for all $i \in [N]$
  $\boldsymbol{V}_t = \arg\min_{\boldsymbol{v}\in\mathcal{S}} \boldsymbol{v}^\intercal \left(\eta_t \widehat{\boldsymbol{L}}_{t-1} - \boldsymbol{Z}_t\right)$
  Receive loss $\boldsymbol{V}_t^\intercal \boldsymbol{\ell}_t$
  Observe graph G_t
  Observe pairs $\{i, \ell_{t,i}\}$ for all i, such that $(j \to i) \in G_t$ and $\boldsymbol{v}(I_t)_j = 1$
  Compute $K_{t,i}$ for all $i \in [N]$ using Eq. (3.16)
  $\hat{\ell}_{t,i} = K_{t,i} O_{t,i} \ell_{t,i}$
end for

5.4 Performance guarantees for FPL-IX

The following theorem states the performance guarantee for FPL-IX in terms of the learning rates and random variables of the form

$$Q_t^{\mathrm{FPL}}(c) = \sum_{i=1}^{N} \frac{q_{t,i}}{o_{t,i} + c}.$$

Note that $Q_t^{\mathrm{FPL}}(c) = Q(m, 0, c)$ where $q_{t,i}$ takes the role of $p_{t,i}$ in the definition of $Q(m, \delta, c)$.

Theorem 11. *Assume $\gamma_t \leq 1/2$ for all t and $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_T$. The regret of* FPL-IX *satisfies*

$$R_T \leq \frac{m\left(\log N + 1\right)}{\eta_T} + 4m \sum_{t=1}^{T} \eta_t \mathbb{E}\left[Q_t^{\mathrm{FPL}}\left(\frac{\gamma_t}{1-\gamma_t}\right)\right] + \sum_{t=1}^{T} \gamma_t \mathbb{E}\left[Q_t^{\mathrm{FPL}}(\gamma_t)\right].$$

The proof of the theorem differs from the proofs for algorithms based on Exp3. Therefore, we present here the key points of the analysis while the complete proof can be found in Section 6.5.

Proof sketch. As usual for analyzing FPL methods [Kalai and Vempala, 2005, Hutter and Poland, 2004, Neu and Bartók, 2013], we first define a hypothetical learner that uses a time-independent perturbation vector $\widetilde{\boldsymbol{Z}} \sim \boldsymbol{Z}_1$ and has access to $\hat{\boldsymbol{\ell}}_t$ on top of $\widehat{\boldsymbol{L}}_{t-1}$

$$\widetilde{\boldsymbol{V}}_t = \operatorname*{arg\,min}_{\boldsymbol{v}\in\mathcal{S}} \boldsymbol{v}^\mathsf{T}\left(\eta_t \widehat{\boldsymbol{L}}_t - \widetilde{\boldsymbol{Z}}\right).$$

Clearly, this learner is infeasible as it uses observations from the future. Also, observe that this learner does not actually interact with the environment and depends on the predictions made by the actual learner only through the loss estimates. By standard arguments, we can prove

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\widetilde{\boldsymbol{V}}_t - \boldsymbol{v}\right)^\mathsf{T}\hat{\boldsymbol{\ell}}_t\right] \le \frac{m\left(\log N + 1\right)}{\eta_T}.$$

Using the techniques of Neu and Bartók [2013], we can relate the performance of $\boldsymbol{V}_t$ to that of $\widetilde{\boldsymbol{V}}_t$, which we can further upper bounded after a long and tedious calculation as

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{V}_t - \widetilde{\boldsymbol{V}}_t)^\mathsf{T}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] &\le \eta_t\,\mathbb{E}\left[\left(\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\hat{\boldsymbol{\ell}}_t\right)^2 \middle| \mathcal{F}_{t-1}\right] \\ &\le 4m\eta_t\mathbb{E}\left[Q_t^{\mathrm{FPL}}\left(\frac{\gamma}{1-\gamma}\right)\middle| \mathcal{F}_{t-1}\right].\end{aligned}$$

The result follows by observing that $\mathbb{E}\left[\boldsymbol{v}^\mathsf{T}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] \le \boldsymbol{v}^\mathsf{T}\boldsymbol{\ell}_t$ for any fixed $\boldsymbol{v} \in \mathcal{S}$ by the optimistic property of the IX estimate and also from the fact that by the definition of the estimates we infer that

$$\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] \ge \mathbb{E}\left[\boldsymbol{V}_t^\mathsf{T}\boldsymbol{\ell}_t \middle| \mathcal{F}_{t-1}\right] - \gamma_t\mathbb{E}\left[Q_t^{\mathrm{FPL}}(\gamma_t)\right].$$

□

The next step is using Lemma 22 to bound the last two expectations in the bound of Theorem 11. Note that $Q_t^{\mathrm{FPL}}(c)$ is in the form of $Q_t(m, 0, c)$. This gives us

$$Q_t^{\mathrm{FPL}}(c) = \sum_{i=1}^{N} \frac{q_{t,i}}{o_{t,i} + c} \leq 2m\alpha_t \log\left(1 + \frac{N^2 + 2Nc}{c\alpha_t}\right) + 2m, \tag{3.18}$$

for all $t \in [T]$ and any $c \in (0, 1)$.

We are now ready to state the main result of this section. It is obtained by combining Theorem 11, the bound on $Q_t^{\mathrm{FPL}}(c)$ given by inequality (3.18), and following variant of Lemma 21

$$\sum_{t=1}^{T} \frac{\alpha_t}{\sqrt{N + \sum_{s=1}^{t-1} \widetilde{\alpha}_s}} \leq \sum_{t=1}^{T} \frac{\alpha_t}{\sqrt{\sum_{s=1}^{t} \alpha_s / C}} \leq 2\sqrt{C \sum_{t=1}^{T} \alpha_t} \leq 2\sqrt{N + C \sum_{t=1}^{T} \alpha_t}.$$

Corollary 5. *Assume that for all $t \in [T]$, $\alpha_t / C \leq \widetilde{\alpha}_t \leq \alpha_t \leq N$ is satisfied for some $C > 1$, and assume $mN > 4$. Setting $\eta_t = \gamma_t = \sqrt{(\log N + 1) \,/\, \left(m\left(N + \sum_{s=1}^{t-1} \widetilde{\alpha}_s\right)\right)}$, the regret of* FPL-IX *satisfies*

$$R_T \leq H m^{3/2} \sqrt{\left(N + C \sum_{t=1}^{T} \alpha_t\right)(\log N + 1)}, \quad \textit{where } H = \mathcal{O}(\log(mNT)).$$

6 Analysis

In this section we provide proofs for all the main theorems, concerning all presented algorithms, in this chapter. The most of the proofs follow an analysis of basic Exp3 algorithm with implicit exploration presented in Section 1.4.

6.1 Regret bound of Exp3-IX

Proof (Theorem 7). The first step of the proof is using a general bound (Lemma 20) which holds for every algorithm based on Exp3. Note that the only condition to be

satisfied is that the sequence of learning rates is non-increasing ($\eta_{t+1} \leq \eta_t$). This holds from the algorithm design. Therefore, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right] \tag{3.19}$$

Now we bound every expectation in the previous expression individually. First, let's look at the first expectation on the left-hand side. Every term of the sum can be controlled as

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\middle|\mathcal{F}_{t-1}\right] &= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} + \sum_{i=1}^{N} p_{t,i}\ell_{t,i}\left(\frac{o_{t,i}}{o_{t,i}+\gamma_t} - 1\right) \\
&= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \sum_{i=1}^{N} p_{t,i}\ell_{t,i}\left(\frac{\gamma_t}{o_{t,i}+\gamma_t}\right) \\
&\geq \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t Q_t^{\mathrm{IX}}.
\end{aligned}$$

Next step is bounding the last expectation on the right-hand side of equation 3.19. Every single term in the sum can be bounded as

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\middle|\mathcal{F}_{t-1}\right] &= \sum_{i=1}^{N} p_{t,i}\frac{\ell_{t,i}^2}{(o_{t,i}+\gamma_t)^2}o_{t,i} \\
&\leq \sum_{i=1}^{N} p_{t,i}\frac{\ell_{t,i}^2}{(o_{t,i}+\gamma_t)o_{t,i}}o_{t,i} \\
&\leq \sum_{i=1}^{N} p_{t,i}\frac{1}{(o_{t,i}+\gamma_t)o_{t,i}}o_{t,i} \\
&= \sum_{i=1}^{N}\frac{p_{t,i}}{o_{t,i}+\gamma_t} = Q_t^{\mathrm{IX}}.
\end{aligned}$$

Combining these bounds yields

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{\eta_t}{2}+\gamma_t\right)Q_t^{\mathrm{IX}}\right].$$

To proceed, we substitute the parameter choice $\eta_t = \sqrt{(\log N)/2(N + \sum_{s=1}^{t-1} Q_s^{\mathrm{IX}})}$ and $\gamma_t = \eta_t/2$. First term on the right-hand side give us

$$\mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] = \mathbb{E}\left[\sqrt{2(\log N)\left(N + \textstyle\sum_{t=1}^{T} Q_t^{\mathrm{IX}}\right)}\right]$$

For the second therm we use the fact that $Q_t^{\mathrm{IX}} \leq N$ together with Lemma 21 to get

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\left(\frac{\eta_t}{2} + \gamma_t\right) Q_t^{\mathrm{IX}}\right] &= \mathbb{E}\left[\sqrt{\frac{\log N}{2}} \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX}}}{\sqrt{N + \sum_{s=1}^{t-1} Q_s^{\mathrm{IX}}}}\right] \\
&\leq \mathbb{E}\left[\sqrt{\frac{\log N}{2}} \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX}}}{\sqrt{\sum_{s=1}^{t} Q_s^{\mathrm{IX}}}}\right] \\
&\leq \mathbb{E}\left[\sqrt{2\log N\left(\textstyle\sum_{t=1}^{T} Q_t^{\mathrm{IX}}\right)}\right] \\
&\leq \mathbb{E}\left[\sqrt{2\log N\left(N + \textstyle\sum_{t=1}^{T} Q_t^{\mathrm{IX}}\right)}\right]
\end{aligned}$$

Using these two bounds we get following inequality:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\sqrt{8(\log N)\left(N + \textstyle\sum_{t=1}^{T} Q_t^{\mathrm{IX}}\right)}\right],$$

which together with the fact that our estimates $\widehat{L}_{T,j}$ are optimistic for all T and j (thanks to the implicit exploration) concludes the proof of the theorem. □

6.2 Regret bound of Exp3-Res

Proof (Theorem 8). Similarly like Exp3-IX algorithm, Exp3-Res algorithm is based on Exp3 and follows an algorithm template described in Algorithm 4. Moreover, adaptive learning rates in Exp3-Res are non-increasing ($\eta_{t+1} \leq \eta_t$). This enables us

to use Lemma 20 for the first part of the analysis and get following inequality

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\eta_t \sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right] \tag{3.20}$$

which holds for any $j \in [N]$. The goal of the second part of the analysis is to construct bounds for each of the expectations in the previous inequality. For the first term on the left-hand side, we use Lemma 26 to get the lower-bound

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] \geq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\ell_{t,i}\right] + \sqrt{T}.$$

Note that this is the only step in the analysis where the actual magnitude (and not just the sign) of the bias of the loss estimates shows up. Anything bigger than $\sqrt{T}$ would degrade our final regret bound.

The second term on the left-hand side can be lower bounded simply as

$$-\mathbb{E}\left[\widehat{L}_{T,j}\right] \geq -\mathbb{E}\left[L_{T,j}\right]$$

since our estimates are optimistic. We are left with bounding the two terms on the right-hand side. To simplify some notation below, let us define $b_t = \sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2$. For the first term, we use the definition of η_t to get

$$\mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] = \mathbb{E}\left[\sqrt{\left(N^2 + \sum_{t=1}^{T} b_t\right)\log N}\right] \leq \sqrt{\left(N^2 + \sum_{t=1}^{T}\mathbb{E}\left[b_t\right]\right)\log N}.$$

The last inequality follows from Jensen's inequality. Now we bound the second term on the right-hand side. By our definition of η_t and the help of Lemma 21, we can

bound it as

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t b_t}{2}\right] &= \mathbb{E}\left[\sum_{t=1}^{T}\frac{b_t\sqrt{\log N}}{2\sqrt{N^2+\sum_{s=1}^{t-1}b_s}}\right]\\ &\leq \mathbb{E}\left[\sqrt{\left(N^2+\sum_{t=1}^{T}b_t\right)\log N}\right]\\ &\leq \sqrt{\left(N^2+\sum_{t=1}^{T}\mathbb{E}\left[b_t\right]\right)\log N},\end{aligned}$$

where we also used the fact that $N^2 \geq b_t$ and Jensen's inequality in the last line. Therefore, whole right hand side of equation 3.20 can be bounded by $2\sqrt{\left(N^2+\sum_{t=1}^{T}\mathbb{E}\left[b_t\right]\right)\log N}$. We continue by bounding $\mathbb{E}\left[b_t\right]$:

$$\begin{aligned}\mathbb{E}_t\left[\sum_{i=1}^{N}p_{t,i}(\hat{\ell}_{t,i})^2\right] &= \sum_{i=1}^{N}p_{t,i}\ell_{t,i}^2\mathbb{E}_t\left[O_{t,i}G_{t,i}^2\right]\\ &\leq \sum_{i=1}^{N}p_{t,i}o_{t,i}\frac{2-o_{t,i}}{o_{t,i}^2} \leq \frac{2}{r_t},\end{aligned} \tag{3.21}$$

where we used $o_{t,i} \geq r_t$ together with the second part of Lemma 25 which gives us

$$\begin{aligned}\mathbb{E}_t\left[G_{t,i}^2\right] &= \frac{2-o_{t,i}}{o_{t,i}^2}+\frac{1}{o_{t,i}^2}(1-o_{t,i})^{N-2}\Big(o_{t,i}^2+o_{t,i}-2+2o_{t,i}(N-2)(o-1)\Big)\\ &\leq \frac{2-o_{t,i}}{o_{t,i}^2},\end{aligned}$$

since both $o_{t,i}^2+o_{t,i}-2$ and $2o_{t,i}(N-2)(o-1)$ are non-positive. Thus, we obtain

$$\mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t b_t}{2}\right] \leq 2\sqrt{\left(\sum_{t=1}^{T}\frac{1}{r_t}+N^2\right)\log N}. \tag{3.22}$$

Finally, combining everything together, we obtain the regret bound

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}p_{t,i}\ell_{t,i}\right]-\min_{j\in[N]}\mathbb{E}\left[\sum_{t=1}^{T}\ell_{t,j}\right] \leq 2\sqrt{\left(N^2+\sum_{t=1}^{T}\frac{1}{r_t}\right)\log N}+\sqrt{T}.$$

□

6.3 Regret bound of EXP3-IXT

Proof (Theorem 9). Recall that we set $s_{j,i}$ to 0 if there is no edge (and therefore weight) from j to i and that Q_t^{IXT} is defined as

$$Q_t^{\text{IXT}} = \sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i} + \sum_{j \neq i} p_{t,j} s_{j,i} \mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t}$$

Note that Q_t^{IXT} is defined as $Q(1, 0, \gamma_t)$ corresponding to the graph G_t after thresholding by ε_t. EXP3-IXT is based on EXP3 template (Algorithm 4) and therefore, the starting point of our analysis is inequality in Lemma 20 which applies to EXP3-IXT as well.

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right] \tag{3.23}$$

For the first expectation on the left-hand side of the previous inequality we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\middle|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{c_{t,i}\mathbb{1}\{s_{I_t,i} \geq \varepsilon_t\}}{\sum_{j\neq i} p_{t,j}s_{j,i}\mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t}\middle|\mathcal{F}_{t-1}\right] \\
&= \sum_{i=1}^{N} p_{t,i}\frac{\sum_{j\neq i} p_{t,j}s_{j,i}\mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\}\ell_{t,i}}{\sum_{j\neq i} p_{t,j}s_{j,i}\mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t} \\
&\geq \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t \sum_{i=1}^{N}\frac{p_{t,i}}{\sum_{j\neq i} p_{t,j}s_{j,i}\mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t} \\
&= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t Q_t^{\text{IXT}}.
\end{aligned}$$

Furthermore, the last expectation on the right-hand side of equation 3.23 can be

bounded as

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2 \middle| \mathcal{F}_{t-1}\right] \\
&\quad= \sum_{i=1}^{N} p_{t,i} \frac{\mathbb{E}\left[s_{I_t,i}^2 \mathbb{1}\left\{s_{I_t,i} \geq \varepsilon_t\right\} \middle| \mathcal{F}_{t-1}\right] \ell_{t,i}^2 + \mathbb{E}\left[\left(1 - s_{I_t,i}\mathbb{1}\left\{s_{I_t,i} \geq \varepsilon_t\right\}\right)^2 \middle| \mathcal{F}_{t-1}\right] \mathbb{E}\left[\xi_{t,i}^2 \middle| \mathcal{F}_{t-1}\right]}{\left(\sum_{j\neq i} p_{t,j} s_{j,i} \mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t\right)^2} \\
&\quad\leq \sum_{i=1}^{N} p_{t,i} \frac{\sum_{j\in N_i^-} p_{t,j} s_{j,i}^2 + R^2}{\left(\sum_{j\neq i} p_{t,j} s_{j,i} \mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t\right)^2} \\
&\quad\leq \frac{1}{\varepsilon_t} \sum_{i=1}^{N} p_{t,i} \frac{1+R^2}{\sum_{j\neq i} p_{t,j} s_{j,i} \mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t} = \frac{(1+R^2)}{\varepsilon_t} Q_t^{\mathrm{IX_T}},
\end{aligned}$$

where the last inequality uses that $\sum_{j\neq i} p_{t,j} s_{j,i} \mathbb{1}\left\{s_{j,i} \geq \varepsilon_t\right\} + \gamma_t \geq \varepsilon_t$.

To finish the proof of the theorem, we use previous bounds, the fact that our loss estimates are optimistic ($\mathbb{E}[\widehat{L}_{t,j}] < L_{t,j}$), and Lemma 21 with

$$\eta_t = \sqrt{\frac{\log N}{2(1+R^2)\left(\frac{N}{\varepsilon_t} + \sum_{s=1}^{t-1} \frac{Q_s^{\mathrm{IX_T}}}{\varepsilon_s}\right)}}, \qquad \gamma_t = \frac{1+R^2}{2\varepsilon_t}\eta_t.$$

This gives us

$$\begin{aligned}
R_T &\leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}} + \sum_{t=1}^{T}\left(\gamma_t + \frac{(1+R^2)\eta_t}{2\varepsilon_t}\right) Q_t^{\mathrm{IX_T}}\right] \\
&= \mathbb{E}\left[\frac{\log N}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}(1+R^2)\eta_t\right] \\
&\leq \mathbb{E}\left[\sqrt{2(1+R^2)(\log N)}\left(\sqrt{\left(N + \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}\right)} + \frac{1}{2}\sum_{t=1}^{T} \frac{\frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}}{\sqrt{N + \sum_{s=1}^{t-1} \frac{Q_s^{\mathrm{IX_T}}}{\varepsilon_s}}}\right)\right] \\
&\leq \mathbb{E}\left[\sqrt{2(1+R^2)(\log N)}\left(\sqrt{\left(N + \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}\right)} + \frac{1}{2}\sum_{t=1}^{T} \frac{\frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}}{\sqrt{\sum_{s=1}^{t} \frac{Q_s^{\mathrm{IX_T}}}{\varepsilon_s}}}\right)\right] \\
&\leq \mathbb{E}\left[\sqrt{8(1+R^2)(\log N)\left(N + \sum_{t=1}^{T} \frac{Q_t^{\mathrm{IX_T}}}{\varepsilon_t}\right)}\right]
\end{aligned}$$

where we set ε_{T+1} to 1 and use the fact that Q_t^{IXT} can be upper bounded by N. □

6.4 Regret bound of Exp3-WIX

Proof (Theorem 10). In principle, our analysis combines Lemma 20, standard tools for analyzing Exp3 with adaptive learning rates, and ideas from Alon et al. [2013] and the analysis of Exp3-IX, while also heavily exploiting the structure of our loss estimates (3.13). In particular, these estimates allow us to bound the expected regret of Exp3-WIX in terms of the quantities $(Q_t^{\text{WIX}})_t$ for graph G_t defined as

$$Q_t^{\text{WIX}} = \sum_{i=1}^{N} \frac{p_{t,i}}{p_{t,i} + \sum_{j \neq i} p_{t,j} s_{j,i}^2 + \gamma_t}.$$

Note that $Q^{\text{WIX}} = Q(1, 1, \gamma_t)$.

Since Exp3-WIX is based on Exp3, i.e. follows Algorithm 4, and sequence $(\eta_t)_t$ is non-increasing, we can use Lemma 20 for the first part of the analysis to get

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\right] - \mathbb{E}\left[\widehat{L}_{T,j}\right] \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\frac{\eta_t}{2}\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\right]. \tag{3.24}$$

For the following part of the analysis, we use a slightly more general form of our loss estimates with general power δ of the weights. We use this definition to bring an insight to the choice of our loss estimates (we use $\delta = 1$ in our loss estimates).

$$\hat{\ell}_{t,i} = \sum_{i=1}^{N} p_{t,i} \frac{s_{I_t,i}^{\delta} c_{t,i}}{p_{t,i} + \sum_{j \neq i} p_{t,j} s_{j,i}^{1+\delta} + \gamma_t} = \sum_{i=1}^{N} p_{t,i} \frac{s_{I_t,i}^{1+\delta}\ell_{t,i} + s_{I_t,i}^{\delta}(1 - s_{I_t,i})\xi_{t,i}}{p_{t,i} + \sum_{j \neq i} p_{t,j} s_{j,i}^{1+\delta} + \gamma_t}.$$

Later we show that $\delta = 1$ is optimal, which recovers the loss estimates (3.13). The next step is to bound individual expectations in bound (3.24). For the first expecta-

tion on the left-hand side we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\hat{\ell}_{t,i}\middle|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}\frac{s^{\delta}_{I_t,i}c_{t,i}}{p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t}\middle|\mathcal{F}_{t-1}\right] \\
&= \sum_{i=1}^{N} p_{t,i}\frac{\sum_{j=1}^{N}p_{t,j}s_{j,i}^{1+\delta}\ell_{t,i}}{p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t} \\
&\geq \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t\sum_{i=1}^{N}\frac{p_{t,i}}{p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t} \\
&= \sum_{i=1}^{N} p_{t,i}\ell_{t,i} - \gamma_t Q_t(1,\delta,\gamma_t).
\end{aligned}$$

For the second expectation on the right-hand side of (3.24) we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{N} p_{t,i}(\hat{\ell}_{t,i})^2\middle|\mathcal{F}_{t-1}\right] \\
&\quad= \sum_{i=1}^{N} p_{t,i}\frac{\mathbb{E}\left[s_{I_t,i}^{2+2\delta}\middle|\mathcal{F}_{t-1}\right]\ell_{t,i}^2 + \mathbb{E}\left[s_{I_t,i}^{2\delta}(1-s_{I_t,i})^2\middle|\mathcal{F}_{t-1}\right]\mathbb{E}\left[\xi_{t,i}^2\middle|\mathcal{F}_{t-1}\right]}{\left(p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t\right)^2} \\
&\quad\leq \sum_{i=1}^{N} p_{t,i}\frac{\sum_{j=1}^{N}p_{t,j}s_{j,i}^{2+2\delta}+\sum_{j=1}^{N}p_{t,j}s_{j,i}^{2\delta}R^2}{\left(p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t\right)^2} \\
&\quad\leq \sum_{i=1}^{N} p_{t,i}\frac{1+R^2}{p_{t,i}+\sum_{j\neq i}p_{t,j}s_{j,i}^{1+\delta}+\gamma_t} = (1+R^2)Q_t(1,\delta,\gamma_t).
\end{aligned}$$

Where the last inequality holds for $\delta \geq 1$. Note that $Q(1,\delta,\gamma_t)$ is a non-decreasing function in δ and therefore, we set $\delta = 1$ to optimize the bound.

Remark 10. *Since* Exp3-IXt *algorithm uses* $\delta = 0$*, we can not use the previous bound in the analysis of* Exp3-IXt *algorithm. This is also the reason for using thresholding in order to work around this problem.*

For the second expectation on the left-hand side of (3.24) we use the fact that our loss estimates are negatively biased and therefore $-\widehat{L}_{t,i}$ can be lower-bounded simply

by $-L_{t,i}$. Applying these bounds to (3.24) we obtain

$$R_T \leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}} + \sum_{t=1}^{T}\left(\gamma_t + \frac{(1+R^2)\eta_t}{2}\right)Q_t^{\mathrm{WIX}}\right].$$

Using Lemma 21 together with the definition of γ_t and η_t we get

$$\begin{aligned}
R_T &\leq \mathbb{E}\left[\frac{\log N}{\eta_{T+1}} + \sum_{t=1}^{T}\left(1+R^2\right)\eta_t Q_t^{\mathrm{WIX}}\right] \\
&\leq \mathbb{E}\left[\sqrt{2(1+R^2)(\log N)}\left(\sqrt{\left(N+\textstyle\sum_{t=1}^{T}Q_t^{\mathrm{WIX}}\right)} + \frac{1}{2}\sum_{t=1}^{T}\frac{Q_t^{\mathrm{WIX}}}{\sqrt{N+\sum_{s=1}^{t-1}Q_s^{\mathrm{WIX}}}}\right)\right] \\
&\leq \mathbb{E}\left[\sqrt{2(1+R^2)(\log N)}\left(\sqrt{\left(N+\textstyle\sum_{t=1}^{T}Q_t^{\mathrm{WIX}}\right)} + \frac{1}{2}\sum_{t=1}^{T}\frac{Q_t^{\mathrm{WIX}}}{\sqrt{\sum_{s=1}^{t}Q_s^{\mathrm{WIX}}}}\right)\right] \\
&\leq \mathbb{E}\left[\sqrt{8(1+R^2)(\log N)\left(N+\textstyle\sum_{t=1}^{T}Q_t^{\mathrm{WIX}}\right)}\right].
\end{aligned}$$

This concludes the proof of the theorem. □

6.5 Regret bound of FPL-IX

The analysis of FPL-IX algorithm combines some techniques used by Kalai and Vempala [2005], Hutter and Poland [2004], and Neu and Bartók [2013] for analyzing FPL-style learners. Our proofs also heavily rely on some specific properties of the IX loss estimate defined in Equation 3.17. The most important difference from the analysis presented in Section 6.1 is that now we are not able to use random learning rates as we cannot compute the values corresponding to Q_t efficiently. In fact, these values are observable in the information-theoretic sense, so we could prove bounds similar to Theorem 7 had we had access to infinite computational resources. As our focus is on computationally efficient algorithms, we choose to pursue a different path. In particular, our learning rates will be tuned according to efficiently computable

approximations $\widetilde{\alpha}_t$ of the respective independence numbers α_t that satisfy $\alpha_t/C \leq \widetilde{\alpha}_t \leq \alpha_t \leq N$ for some $C \geq 1$. For the sake of simplicity, we analyze the algorithm in the oblivious adversary model.

We begin with a statement that concerns the performance of the imaginary learner that predicts $\widetilde{\boldsymbol{V}}_t$ in round t.

Lemma 27. *Assume $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_T$. For any sequence of loss estimates, the expected regret of the hypothetical learner against any fixed action $\boldsymbol{v} \in \mathcal{S}$ satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\widetilde{\boldsymbol{V}}_t - \boldsymbol{v}\right)^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t\right] \leq \frac{m\left(\log N + 1\right)}{\eta_T}.$$

Proof. For simplicity, define $\beta_t = 1/\eta_t$ for $t \geq 1$ and $\beta_0 = 0$. We start by applying the classical follow-the-leader/be-the-leader lemma (see, e.g., [Cesa-Bianchi and Lugosi, 2006, Lemma 3.1]) to the loss sequence defined as $(\hat{\boldsymbol{\ell}}_1 - \widetilde{\boldsymbol{Z}}\beta_1, \hat{\boldsymbol{\ell}}_2 - \widetilde{\boldsymbol{Z}}(\beta_2 - \beta_1), \ldots, \hat{\boldsymbol{\ell}}_T - \widetilde{\boldsymbol{Z}}(\beta_T - \beta_{T-1}))$ to obtain

$$\sum_{t=1}^{T}\widetilde{\boldsymbol{V}}_t^{\mathsf{T}}\left(\hat{\boldsymbol{\ell}}_t - \widetilde{\boldsymbol{Z}}\left(\beta_t - \beta_{t-1}\right)\right) \leq \widetilde{\boldsymbol{V}}_T^{\mathsf{T}}\left(\widehat{\boldsymbol{L}}_T - \widetilde{\boldsymbol{Z}}\beta_T\right) \leq \boldsymbol{v}^{\mathsf{T}}\left(\widehat{\boldsymbol{L}}_T - \widetilde{\boldsymbol{Z}}\beta_T\right).$$

After reordering and observing that $-\boldsymbol{v}^{\mathsf{T}}\widetilde{\boldsymbol{Z}} \leq 0$, we get

$$\begin{aligned}\sum_{t=1}^{T}\left(\widetilde{\boldsymbol{V}}_t - \boldsymbol{v}\right)^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t &\leq \sum_{t=1}^{T}(\beta_t - \beta_{t-1})\widetilde{\boldsymbol{V}}_t^{\mathsf{T}}\widetilde{\boldsymbol{Z}} \\ &\leq \left\|\widetilde{\boldsymbol{V}}_t\right\|_1\left\|\widetilde{\boldsymbol{Z}}\right\|_\infty\sum_{t=1}^{T}(\beta_t - \beta_{t-1}) = \left\|\widetilde{\boldsymbol{V}}_t\right\|_1\left\|\widetilde{\boldsymbol{Z}}\right\|_\infty\beta_T.\end{aligned}$$

The result follows from using our uniform upper bound on $\|\boldsymbol{v}\|_1$ for all $\boldsymbol{v}$ and the well-known bound $\mathbb{E}\left[\left\|\widetilde{\boldsymbol{Z}}\right\|_\infty\right] \leq \log d + 1$. □

The following result can be extracted from the proof of Theorem 1 of Neu and Bartók [2013].

Lemma 28. *For any sequence of nonnegative loss estimates,*

$$\mathbb{E}\left[(\widetilde{\boldsymbol{V}}_{t-1} - \widetilde{\boldsymbol{V}}_t)^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_t\right] \leq \eta_t\,\mathbb{E}\left[\left(\widetilde{\boldsymbol{V}}_{t-1}^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t\right)^2 \middle| \mathcal{F}_t\right].$$

Using these two lemmas, we can prove the following lemma that upper bounds the total expected regret of FPL-IX in terms of the sum of the variables

$$Q_t^{\mathrm{FPL}}(c) = \sum_{i=1}^{N} \frac{q_{t,i}}{o_{t,i} + c}.$$

Lemma 29. *Assume that $\gamma_t \leq 1/2$ for all t. Then,*

$$\sum_{t=1}^{T} \mathbb{E}\left[\boldsymbol{V}_t^{\intercal}\boldsymbol{\ell}_t \middle| \mathcal{F}_{t-1}\right] \leq \sum_{t=1}^{T} \mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^{\intercal}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] + \\ + 4m \sum_{t=1}^{T} \eta_t \mathbb{E}\left[Q_t^{\mathrm{FPL}}\left(\frac{\gamma_t}{1-\gamma_t}\right)\right] + \sum_{t=1}^{T} \gamma_t \mathbb{E}\left[Q_t^{\mathrm{FPL}}(\gamma_t)\right].$$

Proof. First, note that Lemma 28 implies

$$\mathbb{E}\left[(\widetilde{\boldsymbol{V}}_{t-1} - \widetilde{\boldsymbol{V}}_t)^{\intercal}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] \leq \eta_t \, \mathbb{E}\left[\left(\widetilde{\boldsymbol{V}}_{t-1}^{\intercal}\hat{\boldsymbol{\ell}}_t\right)^2 \middle| \mathcal{F}_{t-1}\right]$$

by the tower rule of expectation. We start by observing that

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^{\intercal}\hat{\boldsymbol{\ell}}_t \middle| \mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\sum_{i=1}^{N} q_{t,i}\hat{\ell}_{t,i} \middle| \mathcal{F}_{t-1}\right] = \mathbb{E}\left[\sum_{i=1}^{N} q_{t,i}\frac{\ell_{t,i}}{o_{t,i} + (1-o_{t,i})\gamma_t} O_{t,i} \middle| \mathcal{F}_{t-1}\right] \\
&\geq \mathbb{E}\left[\sum_{i=1}^{N} q_{t,i}\frac{\ell_{t,i}\left(O_{t,i} + (1-o_{t,i})\gamma_t\right)}{o_{t,i} + (1-o_{t,i})\gamma_t} - \gamma_t \sum_{i=1}^{N} q_{t,i}\frac{1-o_{t,i}}{o_{t,i} + (1-o_{t,i})\gamma_t} \middle| \mathcal{F}_{t-1}\right] \\
&\geq \sum_{i=1}^{N} q_{t,i}\ell_{t,i} - \gamma_t \mathbb{E}\left[\sum_{i=1}^{N} \frac{q_{t,i}(1-o_{t,i})}{o_{t,i} + (1-o_{t,i})\gamma_t} \middle| \mathcal{F}_{t-1}\right] \\
&\geq \sum_{i=1}^{N} q_{t,i}\ell_{t,i} - \gamma_t \mathbb{E}\left[\sum_{i=1}^{N} \frac{q_{t,i}}{o_{t,i} + \gamma_t} \middle| \mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{V}_t^{\intercal}\boldsymbol{\ell}_t \middle| \mathcal{F}_{t-1}\right] - \gamma_t Q_t^{\mathrm{FPL}}(\gamma_t).
\end{aligned}$$

To simplify some notation, let us fix a time t and define $\boldsymbol{V} = \widetilde{\boldsymbol{V}}_{t-1}$. We deduce that

$$
\begin{aligned}
&\mathbb{E}\left[\left(\widetilde{\boldsymbol{V}}_{t-1}^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t\right)^2 \middle| \mathcal{F}_{t-1}\right] \\
&\quad= \mathbb{E}\left[\sum_{j=1}^{d}\sum_{k=1}^{d}\left(V_j\hat{\ell}_{t,j}\right)\left(V_k\hat{\ell}_{t,k}\right)\middle| \mathcal{F}_{t-1}\right] \\
&\quad= \mathbb{E}\left[\sum_{j=1}^{d}\sum_{k=1}^{d}\left(V_jK_{t,j}O_{t,j}\ell_{t,j}\right)\left(V_kK_{t,k}O_{t,k}\ell_{t,k}\right)\middle| \mathcal{F}_{t-1}\right] && \text{(def. of } \hat{\boldsymbol{\ell}}_t\text{)} \\
&\quad\le \mathbb{E}\left[\sum_{j=1}^{d}\sum_{k=1}^{d}\frac{K_{t,j}^2+K_{t,k}^2}{2}\left(V_jO_{t,j}\ell_{t,j}\right)\left(V_kO_{t,k}\ell_{t,k}\right)\middle| \mathcal{F}_{t-1}\right] && (2K_{t,j}K_{t,k}\le K_{t,j}^2+K_{t,k}^2) \\
&\quad\le \mathbb{E}\left[\sum_{j=1}^{d}\sum_{k=1}^{d}K_{t,j}^2\left(V_jO_{t,j}\ell_{t,j}\right)\left(V_kO_{t,k}\ell_{t,k}\right)\middle| \mathcal{F}_{t-1}\right] && \text{(symmetry of } j \text{ and } k\text{)} \\
&\quad\le 2\mathbb{E}\left[\sum_{j=1}^{d}\frac{1}{\left(o_{t,j}+(1-o_{t,j})\gamma_t\right)^2}\left(V_jO_{t,j}\ell_{t,j}\right)\sum_{k=1}^{d}V_k\ell_{t,k}\middle| \mathcal{F}_{t-1}\right] \text{(def. of } K_{t,j} \text{ and } O_{t,k}\le 1\text{)} \\
&\quad\le 2m\mathbb{E}\left[\sum_{j=1}^{d}\frac{V_j\ell_{t,j}}{o_{t,j}+(1-o_{t,j})\gamma_t}\middle| \mathcal{F}_{t-1}\right] \\
&\quad\le 2m\sum_{j=1}^{d}\frac{q_{t,j}}{o_{t,j}+(1-o_{t,j})\gamma_t} = \frac{2m}{1-\gamma_t}\sum_{j=1}^{d}\frac{q_{t,j}}{o_{t,j}+\gamma_t/(1-\gamma_t)} \\
&\quad= \frac{2m}{1-\gamma_t}Q_t^{\mathrm{FPL}}\left(\frac{\gamma_t}{1-\gamma_t}\right) \le 4mQ_t^{\mathrm{FPL}}\left(\frac{\gamma_t}{1-\gamma_t}\right),
\end{aligned}
$$

where we used our assumption on γ_t in the last line. The first statement follows from combining the above terms with Lemma 28 and using $\mathbb{E}\left[\boldsymbol{v}^{\mathsf{T}}\hat{\boldsymbol{\ell}}_t\middle|\mathcal{F}_{t-1}\right] \le \boldsymbol{v}^{\mathsf{T}}\boldsymbol{\ell}_t$ by the optimistic property of the loss estimates $\hat{\boldsymbol{\ell}}_t$. □

Chapter 4

Summary and future work

Multi-armed bandit framework is widely used to solve real-word problems where an agent obtains only the feedback of his actions (recommender systems, advertising etc.). However, bandit feedback is sometimes very limited and the problems come with an additional structure. We addressed this issue in this thesis and proposed several extensions of stochastic and adversarial multi-armed bandit problems. These extensions are based on real-word problems and insufficient theoretical and empirical guarantees of existing algorithms.

Spectral bandits for smooth graph functions

The first extension of multi-armed bandit problem we studied is called spectral bandits (Chapter 2). The spectral bandit problem extends the basic stochastic multi-armed bandit setting and is inspired mostly by the applications in recommender systems and targeted advertisement in social networks. In this setting, we are asked to repeatedly maximize an unknown graph function, assumed to be smooth on a given similarity graph. Traditional linear bandits can be applied but their regret scales with the ambient dimension D, either linearly or as a square root, which can be very large.

The main contribution of Chapter 2 is the introduction of a novel quantity called effective dimension, denoted by lower case d, and the introduction of three algorithms, SpectralUCB, SpectralTS, and SpectralEliminator. The effective dimension characterizes the difficulty of the problem: we showed a regret lower bound for the setting which scales with this quantity and regret bounds of the previously mentioned algorithms scale with effective dimension d as well. The benefit of the effective dimension lies in the fact that the effective dimension d is typically much smaller than D for real-world graphs. We also performed experiments and showed

that spectral algorithms are able to leverage the structure of the problem when the reward function is smooth on the graph much better than their linear counterparts.

Future work

In this section, we discuss several open questions concerning spectral bandits.

One of the limitations of the spectral bandit setting is the assumption that the graph is fixed over time. This presents a problem in applications like recommender systems since preferences of users can change over time: some of the movies can become less or more popular and therefore, the links in the underlying graph can change as well. Another similar problem is an introduction of new actions (new movies, new ads etc.). Both of these problems would require to do the eigendecomposition of the graph Laplacian again and recompute reward estimates and confidence bounds again from scratch. This could present a practically intractable problem. Therefore, the open question is whether there is an approach which can deal with changing structure of the problem.

The regret bounds of SpectralTS and SpectralUCB scale linearly with d while SpectralEliminator is the only presented algorithm with upper bound scaling with the root of d and thus matching lower bound. However, the downside of SpectralEliminator is that its empirical performance is poor compared to the other algorithms. Therefore, the open question is whether SpectralTS and SpectralUCB have upper bound scaling with square root of d or whether there is an empirically successful algorithm with an bound scaling with root of d.

Adversarial bandits with side observations

Spectral bandits gain additional information about other actions implicitly by assuming that the reward function is smooth and thus, connected rewards tend to be similar. On the other hand, bandits with side observations take a different, more explicit, approach to obtain additional information. In bandits with side observations, the underlying graph gives us access to other losses simply by revealing the losses (possibly noisy) of all the neighbors of the selected action. One of the main contributions of Chapter 3 is a novel approach to encourage additional exploration of an algorithm. We call it "implicit exploration" and it proved to be computationally

less expensive than the algorithms using mixing. Another contribution of the chapter is formalizing several variants of bandits with side observations, suited for different real-world problems, and providing solutions in the form of algorithms with strong theoretical guarantees. In the next part, we discuss specific contributions in these variants of bandits with side observations.

Adversarial bandits with adversarial side observations

This setting is the same as the partial observability model of Mannor and Shamir [2011], Alon et al. [2013]. The main contribution is the first algorithm, called Exp3-IX, solving the problem without knowing the graph structure beforehand. This algorithm is also the first algorithm using implicit exploration, presented in this thesis. The regret bound of Exp3-IX is of $\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \alpha_t}\right)$ (Corollary 2) which is near-optimal bound for the setting.

Adversarial bandits with stochastic side observations

In this setting, we considered multi-armed bandit problems with stochastic side observations modeled by Erdős–Rényi graphs. Our contribution is a computationally efficient algorithm that operates under the assumption $r_t \geq \log T/(2N-2)$, which essentially guarantees that at least one piece of side observation is generated in every round, with high probability. In this case, our algorithm guarantees a regret bound of $\mathcal{O}\left(\sqrt{\log N \sum_{t=1}^{T} \frac{1}{r_t}}\right)$ (Theorem 8).

The most obvious question is whether it is possible to remove our assumptions on the values of r_t in this setting. We can only give a definite answer in the simple case when all r_t's are identical: In this case, one can think of simply computing the empirical frequency $\widehat{r}_t$ of all previous side observations in round t to estimate the constant r, plug the result into (3.6), and then use the resulting loss estimates in an exponential-weighting scheme. It is relatively straightforward to show that the resulting algorithm satisfies a regret bound of $\widetilde{\mathcal{O}}\left(\sqrt{T/r}\right)$ for all possible values of r, thanks to the fact that $\hat{r}_t$ quickly concentrates around the true value of r. Notice however that this approach clearly breaks down if the r_t's change over time.

In the case of changing r_t's, the number of observations we can use to estimate

r_t is severely limited, so much that we cannot expect any direct estimate of r_t to concentrate around the true value. Our algorithm proposed in Section 3 gets around this problem by directly estimating the importance weights $1/o_{t,i}$ instead of r_t, which enables us to construct reliable loss estimates, although only at the price of our assumption on the range of r_t. While we acknowledge that this assumption can be difficult to confirm a priori in practice, we remark that we find it quite surprising that *any algorithm whatsoever* can take advantage of such limited observations, even under such a restriction. We also point out that for values of r_t that are consistently below our bound, it is not possible to substantially improve the regret bounds of Exp3 which are of $\widetilde{\mathcal{O}}\left(\sqrt{TN}\right)$, as shown by the lower bounds of Alon et al. [2013]. We expect that in several practical applications, one can verify whether the r_t's satisfy our assumption or not, and decide to use Exp3-Res or Exp3 accordingly. In fact, our experiments suggest that our algorithm performs well even if neither of these two assumptions is verified: we have seen that the empirical performance of Exp3-Res is only slightly worse than that of Exp3 even when the values of r_t are very small (Section 3.3). Still, finding out whether our restriction on r_t can be relaxed in general is a very important and interesting question left for future study.

An important corollary of our results is that, under some assumptions, it is possible to leverage side observations in a non-trivial way without having access to the second neighborhoods in the side-observation graphs as defined by Mannor and Shamir [2011]. This complements the recent results of Cohen et al. [2016], who show that non-stochastic side-observations may provide a non-trivial advantage over bandit feedback when the losses are stochastic even when the side-observation graphs are unobserved, but learning with unobserved feedback graphs can be as hard as learning with bandit feedback when both the losses and the graphs are generated by an adversary. A natural question that our work leads to is whether it is possible to efficiently leverage side-observations under significantly weaker assumptions on the observation model.

Adversarial bandits with noisy side observations

The main contribution for this setting is introducing a new partial-observability model for adversarial online learning and proposing two algorithms, Exp3-IXt and Exp3-WIX, with rigorous performance guarantees for this setting. Our regret bounds depend on a newly introduced graph property that we call the effective independence number α^* and the regret bounds for both algorithms are of

$\mathcal{O}\left(\sqrt{\sum_{t=1}^{T}\alpha_t^*}\right)$ (Corollaries 3 and 4). Moreover, Exp3-WIX achieves this bound without any additional assumption while Exp3-IXt needs to know graph beforehand to choose optimal thresholding parameter.

While the recent results of Wu et al. [2015] suggest that the bounds of Exp3-IXt and Exp3-WIX for the setting with noisy side observations are minimax optimal in some special cases of our framework, it is not yet known whether the effective independence number is the exact quantity that characterizes the minimax regret in general—this exciting question remains open for future investigation.

Combinatorial semi-bandit problem with adversarial side observations

In this setting, we presented a computationally efficient algorithm, called FPL-IX, which achieves regret of $\mathcal{O}\left(m^{3/2}\sqrt{(\log(N))\sum_{t=1}^{T}\alpha_t}\right)$.

However, it is known that the minimax optimal regret of combinatorial semi-bandits scales with $\sqrt{m}$. This can be achieved using Online Stochastic Mirror Descent (OSMD) [Audibert et al., 2014] algorithm for the cost of not being efficiently implementable in general. Therefore, we used an FPL-based algorithm which is efficient for the price of the extra square root of m in the regret bound. It is still an open question whether FPL-IX scales linearly with m or whether there is an algorithm with minimax optimal regret while still implementable efficiently.

Another important open problem is, whether there is an efficient algorithm for full bandit feedback where instead of the individual losses of the played arms, the learner observes only their sum.

Bibliography

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Neural Information Processing Systems*, 2011. (→ pages 6, 35, 39, 40, and 46.)

J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: an efficient algorithm for bandit linear optimization. In *Conference on Learning Theory*, 2008. (→ page 6.)

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 2013. (→ pages 6, 9, 16, 30, 40, 42, 45, 46, and 47.)

C. Allenberg, P. Auer, L. Györfi, and G. Ottucsák. Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *Algorithmic Learning Theory*, 2006. (→ page 89.)

N. Alon, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. From bandits to experts: A tale of domination and independence. In *Neural Information Processing Systems*, 2013. (→ pages 64, 67, 69, 78, 83, 84, 88, 89, 98, 99, 100, 102, 107, 113, 116, 127, 135, and 136.)

N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren. Online learning with feedback graphs: Beyond bandits. In *Conference on Learning Theory*, 2015. (→ pages 64, 68, 78, and 100.)

J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 2010. (→ pages 69 and 89.)

J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 2014. (→ pages 7, 114, 115, 116, and 137.)

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 2002. (→ pages 6, 9, 16, 47, and 51.)

P. Auer and R. Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 2010. (→ page 32.)

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 2002a. (→ pages 10, 72, and 102.)

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 2002b. (→ pages 26, 68, 69, 71, 81, 89, 90, 91, and 114.)

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 2002c. (→ pages 77, 81, and 82.)

B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Symposium on Theory Of Computing*, 2004. (→ page 2.)

M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. *SIAM Journal on Computing*, 2014. (→ page 3.)

G. Bartók, D. Pál, and C. Szepesvári. Minimax regret of finite partial-monitoring games in stochastic environments. In *Conference on Learning Theory*, 2011. (→ page 100.)

G. Bartók, D. P. Foster, D. Pál, A. Rakhlin, and C. Szepesvári. Partial monitoring-classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 2014. (→ page 100.)

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Conference on Computational Learning Theory*, 2004. (→ pages 8 and 14.)

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 2006. (→ pages 8, 14, 16, and 20.)

A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2011. (→ page 69.)

D. Billsus, M. J. Pazzani, and J. Chen. A learning agent for wireless news access. In *International Conference on Intelligent User Interfaces*, 2000. (→ pages 8 and 15.)

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012. (→ page 69.)

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 2011. (→ page 18.)

S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, 2012. (→ page 6.)

S. Buccapatnam, A. Eryilmaz, and N. B. Shroff. Stochastic bandits with side observations on networks. In *International Conference on Measurement and Modeling of Computer Systems*, 2014. (→ page 89.)

S. Caron, B. Kveton, M. Lelarge, and S. Bhagat. Leveraging side observations in stochastic bandits. In *Conference on Uncertainty in Artificial Intelligence*, 2012. (→ page 89.)

A. Carpentier and M. Valko. Revealing graph bandits for maximizing local influence. In *International Conference on Artificial Intelligence and Statistics*, 2016. (→ page 89.)

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. (→ pages 79, 101, and 130.)

N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 2012. (→ pages 7, 114, 115, and 116.)

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 1997. (→ page 5.)

N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 2005. (→ page 89.)

N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir. Online learning of noisy data with kernels. *Conference on Learning Theory*, 2010. (→ page 100.)

N. Cesa-Bianchi, C. Gentile, and G. Zappella. A gang of bandits. In *Neural Information Processing Systems*, 2013. (→ page 18.)

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Neural Information Processing Systems*, 2011. (→ page 6.)

D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *Conference on Human Factors in Computing Systems*, 2011. (→ pages 8 and 15.)

W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, 2013. (→ page 114.)

W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 2011. (→ page 6.)

A. Cohen, T. Hazan, and T. Koren. Online learning with feedback graphs without the graphs. In *International Conference on Machine Learning*, 2016. (→ pages 65, 87, and 136.)

R. Combes and A. Proutière. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, 2014. (→ page 18.)

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, 2008. (→ page 6.)

T. Desautels, A. Krause, and J. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. In *International Conference on Machine Learning*, 2012. (→ page 33.)

L. Devroye, G. Lugosi, and G. Neu. Prediction by random-walk perturbation. In *Conference on Learning Theory*, 2013. (→ page 100.)

M. Fang and D. Tao. Networked bandits with disjoint linear payoffs. In *Internattional Conference on Knowledge Discovery and Data Mining*, 2014. (→ page 18.)

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997. (→ pages 5 and 95.)

C. Gentile, S. Li, and G. Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, 2014. (→ page 18.)

Graclus. Graclus, 2013. URL `www.cs.utexas.edu/users/dml/Software/graclus.html`. (→ page 59.)

Q. Gu and J. Han. Online spectral learning on a graph with bandit feedback. In *International Conference on Data Mining*, 2014. (→ page 19.)

L. Györfi and G. Ottucsák. Sequential prediction of unbounded stationary time series. *IEEE Transactions on Information Theory*, 2007. (→ pages 72 and 81.)

A. György, T. Linder, G. Lugosi, and G. Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 2007. (→ page 2.)

M. K. Hanawal, V. Saligrama, M. Valko, and R. Munos. Cheap bandits. In *International Conference on Machine Learning*, 2015. (→ page 19.)

J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the theory of games*, 1957. (→ page 116.)

R. A. Horn and C. R. Johnson. Matrix analysis. Cambridge University Press, 1990.

M. Hutter and J. Poland. Prediction with expert advice by following the perturbed leader for general weights. In *Algorithmic Learning Theory*, 2004. (→ pages 119 and 129.)

M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *ACM conference on Recommender systems*, 2010. (→ page 59.)

D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: An introduction*. Cambridge University Press, 2010. (→ pages 2, 8, and 15.)

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 2005. (→ pages 115, 119, and 129.)

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. In *IEEE International Symposium on Information Theory*, 2009. (→ page 57.)

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Symposium on Theory Of Computing*, 2008. (→ page 18.)

T. Kocák, G. Neu, M. Valko, and R. Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Neural Information Processing Systems*, 2014a. (→ pages 64, 68, 69, 78, 103, and 107.)

T. Kocák, M. Valko, R. Munos, and S. Agrawal. Spectral Thompson sampling. In *AAAI Conference on Artificial Intelligence*, 2014b.

T. Kocák, M. Valko, R. Munos, B. Kveton, and S. Agrawal. Spectral bandits for smooth graph functions with applications in recommender systems. In *AAAI Workshop on Sequential Decision-Making with Big Data*, 2014c.

T. Kocák, G. Neu, and M. Valko. Online learning with noisy side observations. In *International Conference on Artificial Intelligence and Statistics*, 2016a.

T. Kocák, G. Neu, and M. Valko. Online learning with Erdős-Rényi side-observation graphs. In *Conference on Uncertainty in Artificial Intelligence*, 2016b.

W. M. Koolen, M. K. Warmuth, and J. Kivinen. Hedging structured concepts. In *Conference on Learning Theory*, 2010. (→ pages 7, 10, 115, and 116.)

N. Korda, B. Szörényi, and S. Li. Distributed clustering of linear andits in peer to peer networks. In *International Conference on Machine Learning*, 2016. (→ page 18.)

I. Koutis, G. L. Miller, and D. Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding*, 2011. (→ page 33.)

S. Lam and J. Herlocker. MovieLens 1M dataset, 2012. URL `http://www.grouplens.org/node/12`. (→ page 57.)

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International World Wide Web Conference*, 2010. (→ pages 6, 9, 16, 28, and 29.)

S. Li, C. Gentile, A. Karatzoglou, and G. Zappella. Online context-dependent clustering in recommendations based on exploration-exploitation algorithms. *arXiv preprint*, 2015. (→ page 18.)

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 1994. (→ pages 5, 68, and 95.)

U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007. (→ page 20.)

Y. Ma, T.-K. Huang, and J. Schneider. Active search and bandits on graphs using sigma-optimality. In *Conference on Uncertainty in Artificial Intelligence*, 2015. (→ page 19.)

S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *Neural Information Processing Systems*, 2011. (→ pages 10, 64, 65, 67, 69, 78, 87, 98, 99, 100, 101, 113, 135, and 136.)

H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Conference on Learning Theory*, 2004. (→ page 2.)

M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001. (→ pages 8 and 15.)

S. K. Narang, A. Gadde, and A. Ortega. Signal processing techniques for interpolation in graph structured data. In *International Conference on Acoustics, Speech and Signal Processing*, 2013. (→ page 19.)

G. Neu and G. Bartók. An efficient algorithm for learning with semi-bandit feedback. In *Algorithmic Learning Theory*, 2013. (→ pages 91, 115, 116, 117, 119, 129, and 130.)

S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. In *International Conference on Machine Learning*, 2007. (→ page 3.)

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 1952. (→ page 3.)

E. M. Schwartz. *Optimizing adaptive marketing experiments with the multi-armed bandit*. PhD thesis, 2013. (→ page 3.)

Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *International Conference on Machine Learning*, 2014. (→ pages 89 and 91.)

A. Slivkins. Contextual bandits with similarity information. In *Conference on Learning Theory*, 2009. (→ page 18.)

N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010. (→ page 18.)

E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 2003. (→ page 2.)

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933. (→ pages 1, 2, and 6.)

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, 2013. (→ page 18.)

M. Valko, R. Munos, B. Kveton, and T. Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, 2014. (→ pages 21 and 22.)

V. Vovk. Aggregating strategies. In *Proceedings of the third annual workshop on Computational learning theory*, 1990. (→ page 5.)

M. Wainwright. STAT 210B advanced mathematical statistics. *Lecture notes, University of California at Berkeley*, 2015. (→ page 49.)

Y. Wu, A. György, and C. Szepesvári. Online learning with Gaussian payoffs and side observations. In *Neural Information Processing Systems*, 2015. (→ pages 100, 102, 108, and 137.)

J. Y. Yu and S. Mannor. Unimodal bandits. In *International Conference on Machine Learning*, 2011. (→ page 18.)

X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison, 2008. (→ pages 8 and 14.)