Learning with Limited Samples

Chen, Shouyuan

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. Irwin King and Prof. Michael R. Lyu

Thesis/Assessment Committee Members

Professor Shengyu ZHANG (Chair)
Professor Michael R. LYU (Thesis Supervisor)
Professor Irwin KING (Thesis Supervisor)
Professor Lai-Wan CHAN (Committee Member)
Professor Dit-Yan YEUNG (External Examiner)

Abstract of thesis entitled:
Learning with Limited Samples
Submitted by Chen, Shouyuan
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in October 2014

Many machine learning algorithms require a large number of samples as input to achieve good performance. However, in many applications, it is very expensive or simply impossible to collect a large number of samples. These applications require the corresponding algorithms to use only a limited number of samples, which presents an important challenge in the design and analysis of learning algorithms.

In this thesis, we address this challenge by studying several learning problems emerging from the fields of multi-armed bandit (MAB) and tensor completion, which encompass a wide range of applications where the number of available samples is limited. For each of these problems, we identify natural requirements on the number of necessary samples for the corresponding learning problems, and design practical algorithms with provable guarantees that are close to these necessary requirements.

This thesis is divided into four parts. In the first part, we study the combinatorial pure exploration problem in the stochastic MAB setting, where a learner explores a set of arms with the objective of finding the optimal combinatorial structure among many. We present practical learning algorithms, upper bounds and a lower bound for a rich class of combinatorial structures. Our upper and lower bounds show that our algorithms are optimal (within logarithmic factors) in terms of sample complexity in many cases.

In the second part, we study the linear combinatorial bandits problem. We design a learning algorithm for the problem and prove the related regret bound. We apply the algorithm to on-line diversified movie recommendation, where only a limited number of samples are known for each user. Experiment results agree with our theoretical results and demonstrate that our learning algorithm is accurate and sample efficient.

In the third part, we present a fast relative-error approximation algorithm for ridge regression, which is widely used in the learning algorithms for linear bandits. Our experiment results demonstrate the efficiency of our algorithm.

In the final part, we present new algorithms for learning pairwise interaction tensors from a limited number of its entries. We show that, under mild assumptions, it is possible to exactly or robustly learn a pairwise interaction tensor from a nearly minimal number of samples. Our algorithm is the first provable algorithm that guarantees exact recovery of pairwise interaction tensors. The experiments demonstrate promising performance on temporal movie recommendation tasks.

Acknowledgement

I would like to thank my wife, Lijing Qin, who brought me happiness and passion. Her love and support is the key. I would like to thank my parents, without whom none of this would be possible.

I would like to thank my supervisors, Michael R. Lyu and Irwin King, for their advices, encouragements, patience and support. I am deeply indebted to all the time and effort they have devoted towards my research and life here at the Chinese University of Hong Kong.

I would like to thank Wei Chen, Alex Deng and Edward Chang, who mentored me when I was interning in their places. Their viewpoints and intuitions helped to form many important research aspects of this thesis.

I would like to thank Shengyu Zhang, Lai-Wan Chan and Dit-Yan Yeung for being my thesis committee.

I would like to thank my friends in Room 1024: Guang Ling, Qirun Zhang, Chen Cheng, Hongyi Zhang, Tong Zhao, Chao Zhou and Yangfan Zhou. I am grateful to them for all the good times.

To Lijing, my beloved wife.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Many machine learning algorithms require a large number of samples in order to achieve good performances. However, in many applications, it can be too expensive or simply impossible to collect a large number of samples. These applications require the related learning algorithms to use only a limited number of samples.

For example, consider the problem of testing several candidates of a commercial product. The learner often need to find the best candidate by using a limited number of resources (e.g., customer surveys). In medical trials, researchers may wish to find the best drug among many drugs under test, and to treat patients as close as the case when the best drug is known [126]. It is easy to see that collecting treatment results of patients can be a very expensive process. Moreover, in a movie recommender system, an important problem is to predict users' preferences over the movies by using the rating information from users. In this application, the rating information is often very limited, while the learner may be asked to make prediction when a majority of information is missing [17]. Hence, we see that learning effectively from a limited number of samples is a recurring challenge in a diverse applications.

In this thesis, we address this challenge by studying several concrete learning problems emerging from the fields of multi-armed bandit (MAB) and tensor completion. MAB and tensor completion encompass a wide range of applications where the number of available samples is limited. For each of these problems, we identify natural requirements on the number of samples that are necessary for the corresponding learning problems, and design provable and practical algorithms that are close to these necessary requirements. In theory, our results characterize the sample complexity of these learning problems; in practice, our algorithms lead to practical and sample efficient solutions.

The rest of this chapter is organized as follows. In Section 1.2, we review the related work on multi-armed bandit problem including several important extensions. In Section 1.3, we review the literature of matrix completion and tensor completion. In Section 1.4, we briefly describe the learning problems studied in this thesis and summarize our contributions.

1.2 Multi-armed bandits

Multi-armed bandit (MAB) is a classical example of machine learning problems which focus on utilizing a limited number of samples [91, 10, 25]. The original MAB problem is a sequential decision-making problem played by a single player over a sequence of rounds. At each round, the player chooses an action and receives a sample of reward associated with the chosen action. The goal of the player is to maximize her total reward obtained from all the rounds. Historically, the name "multi-armed bandit" is a metaphor: the name "bandit" refers to slot machines; and the MAB problem is a game that, in a casino, a player is facing a number of slot machines and is asked to choose repeatedly slot machines to insert her coins.

The MAB problem was originally motivated by medical trials [126], where an arm corresponds to a drug, and the Bernoulli reward associated with an arm corresponds to the effectiveness of treatment using a certain drug. The objective is to maximize the number of effective treatments, i.e., to perform almost as good as using the best drugs for all patients. Naturally, we do not want apply suboptimal drugs to too many patients (exploitation); while we also want to find out the best drug by experimenting them on the patients (exploration). We can see that an exploration-exploitation dilemma appears naturally in this simple scenario.

Indeed, in order to play the game optimally, the player need to carefully divide her samples (i.e. coins) into (1) exploitation of actions that give good rewards in the past and (2) exploration of actions that may give better rewards in the future. Owing to the universal nature of the trade-off between exploration and exploitation, the MAB model has found applications in diverse domains and attracted considerable attentions.

Most of the existing work of MAB can be divided into two categories by their assumptions on the rewards [25]: *stochastic reward* (where the reward is sampled from a fixed distribution) and *adversarial reward* (where the reward is chosen by an adversarial). Each specific bandit model has a distinct playing strategy: the UCB-1 algorithm for stochastic MABs [91] and the Exp3 algorithm for adversarial MABs [10]. In this thesis, we mainly focus on the stochastic MABs. We refer readers to [25] for a general introduction to MABs, which includes both stochastic and adversarial settings.

The stochastic MAB problem, originally formulated by Robbins [126], can be described as follows. Suppose there are n arms and T rounds. Each arm $i \in [n]$ corresponds to an unknown probability distribution Φ_i on $[0, 1]$. For each round $t = 1, \ldots, T$, the player pulls an arm i_t and observes a reward sampled independently from the corresponding distribution Φ_{i_t}.

In order to evaluate a player's strategy, one can compare her performance with the optimal strategy which always plays the best arm, i.e. the arm with the largest mean reward. The difference between the reward of the optimal strategy and the player's total reward is referred as *cumulative regret* or, in short, *regret*. In the stochastic setting, the regret of a player can be defined as follows. For $i = 1, \ldots, n$, let μ_i denote the expectation of the reward distribution Φ_i. Let $\mu_* = \max_{i \in [n]} \mu_i$. Then, the regret over T rounds is defined as follows

$$R_T = T\mu_* - \sum_{t=1}^{T} \mu_{i_t}.$$

The stochastic MAB problem has been well studied in the literature. The analysis of this problem was pioneered by Lai and Robbins [91], who introduced the upper confidence bound (UCB) techniques for obtaining asymptotic bounds of regret. Auer et al. [9] improved the results of Lai and Robbins [91] and obtained a finite-time regret bound. The resulting UCB1 algorithm guarantees an $O(\log(T))$ regret, which is shown to be essentially optimal.

In the past decade, the MAB problem has been applied into a number of more sophisticated scenarios. Many of these applications are better modeled by extensions of the basic MAB problem. In the following, we investigate three important extensions.

1.2.1 Linear bandits

Stochastic linear bandits are a natural extension of stochastic MABs, where arms are associated with feature vectors and rewards are determined by a linear function. Specifically, in this case, the set of arms $\{1, 2, \ldots, n\}$ is replaced by a compact set $K \subseteq \mathbb{R}^d$. The reward at each round is given by some unknown linear function f defined on K, and the problem is to choose arms from K that is as close as possible to the maximum of the reward function f. Formally, suppose that there are T rounds and, at each round $t = 1, \ldots, T$, the player chooses an arm $x_t \in K$ and receives a reward r_t. In the stochastic linear bandits setting, we assume that $r_t = f(x_t) + \epsilon_t$, where f is an unknown linear function and ϵ_t is a zero-mean random variable. The goal is to maximize the sum of expected rewards $\sum_{t=1}^{T} f(x_t)$, which is equivalent to minimize the cumulative regret.

The linearity assumption on the reward function f occurs naturally in many applications. For example, consider the application of recommending movies to a new user over a sequence of rounds, and the objective is to maximize the user's cumulative ratings on the recommended movies. This application can be naturally formulated as a linear bandit problem as follows. For movie recommendations, it is well-known that the user's ratings over movies can be modeled by a linear function [128, 88, 86, 87]. Specifically, suppose that each movie i is represented by a vector $\mathbf{v}_i \in \mathbb{R}^d$. Then, the user's rating of movie i is modeled as $r_i = \mathbf{u}^T \mathbf{v}_i + \epsilon_i$, where $\mathbf{u}$ is a d-dimensional vector which characterizes the user's preference and ϵ_i is a zero-mean random variable. It is clear that, for a new user who did not rate any movies before, her preference vector $\mathbf{u}$ is unknown. On the other hand, there exist various algorithms based on matrix factorization which can accurately estimate the feature vectors of movies $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ using the rating information of other users [128, 88]. Therefore, we can regard the n movies as n arms with known feature vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. The reward function corresponds to the rating of the user over movies, which is given by $f(\mathbf{v}) = \mathbf{u}^T \mathbf{v}$ for an unknown $\mathbf{u}$. In this case, playing an arm corresponds to recommending a movie to the user, who subsequently discloses the reward by rating the recommended movie.

The stochastic linear bandits problem was introduced by Auer

[8]. His algorithm was later improved by Dani et al. [48], who also gave a near complete characterization to the problem by providing both upper and lower bounds. Li et al. [93] modified the algorithm of Auer [8] by using ridge regression, and applied the resulting algorithm to personalized news recommendation. Abbasi-Yadkori et al. [1] proposed an algorithm similar to Li et al. [93], and provided an analysis using techniques from theory of self-normalized processes. Their analysis improved the regret bound of Dani et al. [48] by a logarithmic factor.

Linear bandits are also studied under the adversarial setting, where the reward function at each round is chosen by an adversary. This setting is also known as online linear optimization with bandit feedback, which was pioneered by Awerbuch and Kleinberg [12] and McMahan and Blum [104]. Dani et al. [47] provided the first optimal regret bound of $O(\sqrt{T})$ based on the Exp2 strategy. In a parallel line of research, Abernethy et al. [2] proposed an algorithm based on online mirror descent with $O(\sqrt{T})$ regret. Their algorithm was further improved by Bubeck et al. [27].

1.2.2 Combinatorial bandits

Combinatorial bandits are a generalization of standard MABs, where, at each round, a subset of arms are played simultaneously. The subset of arms are required to satisfy certain combinatorial constraints and the reward is given by a reward function defined over subsets of arms. There are two variants of combinatorial bandits which have different models of observations: (1) the bandit setting, where the player only observes the reward of the chosen subset; and (2) the semi-bandit setting, where the reward of the chosen subset depends on the rewards of individual arms in the subset and the rewards of these arms are revealed to the player. By using different types of combinatorial constraints and reward functions, combinatorial bandits encompass a wide range of sequential decision-making problems and therefore they have found applications in many domains including online advertisement [43], social influence maximization [43], network routing [90, 62], wireless network [62, 63] and movie recommendation [90].

The general stochastic combinatorial bandit problem was first studied by Gai et al. [62], where the reward functions are restricted to be linear, i.e., the reward of a subset of arms is the sum of rewards of individual arms in the subset. Chen et al. [43] generalized the setting to encompass non-linear reward functions. They proposed the CUCB algorithm and provided tighter regret bounds. Prior to these two work, several specific instances of the general stochastic combinatorial problem have been studied. The problem of playing k arms simultaneously at each round was studied by [5, 37, 97]. Gai et al. [63] studied the matching bandit problem, where arms correspond to edges in a bipartite graph and edges consisting of a matching was played at each round. Liu and Zhao [99] considered a similar settings, where, at each round, a path in a graph was played.

A parallel line of research studies the combinatorial bandit problem in adversarial setting [39, 7, 27]. Most of these work focus on linear reward function. These work are also closely related to the linear bandit problem in the adversarial setting [39, 25]. For non-linear rewards, a few work studied online submodular optimization problems with bandit feedback [143, 118, 72].

1.2.3 Pure exploration bandits

The trade-off between exploration and exploitation is intrinsic in many applications. However, in some application domains, practitioners prefer a dedicated exploration procedure, where the objective is to find the optimal object from a collection of candidates and the reward or loss incurred during exploration can be ignored. The related learning problem, called pure exploration problem of MABs, has received much attention in the literature.

The setting of pure exploration problem of MAB is similar to the setting of standard stochastic MABs, which can be described as follows. There are n arms and each arm i is associated with an unknown reward distribution Φ_i, whose expectation is denoted by μ_i. The game is played over a sequence of rounds. At round t, the player chooses an arm $i_t \in [n]$ and observes a reward sampled independently from the associated reward distribution Φ_{i_t}. In the end of the game, the player is asked to report an arm $o \in [n]$, with the goal of minimizing the *simple regret*, which is defined as $\mu_* - \mu_o$, where $\mu_* = \max_{i \in [n]} \mu_i$ is the largest mean reward. It is easy to see that a zero simple regret means that the player correctly finds the best arm, which means that minimizing simple regret is equivalent to finding the best arm. Bubeck et al. [26] showed that minimizing simple regret is fundamentally different from minimizing cumulative regret. Hence very different algorithms are needed to solve the pure exploration problem.

The pure exploration problem was first studied in a PAC model by Even-Dar et al. [53]. The authors aimed to (1) minimize the number of samples, i.e. the number of rounds, used by the algorithm and (2) guarantee that the simple regret is smaller than ϵ with probability at least $1 - \delta$. They proposed an algorithm called Successive Elimination (SE), which achieves the (ϵ, δ) guarantee with an optimal number of samples (up to logarithmic factors). A matching sample complexity lower bound was proved by Mannor and Tsitsiklis [102] in an earlier work.

Audibert et al. [6] initiated the study of pure exploration problem in the fixed budget setting, where the game is played for at most T rounds and the player needs to minimize the probability of error, which is formally $\Pr[\mu_* - \mu_o > \epsilon]$. They proposed the Successive Rejection (SR) algorithm and analyze its probability of error. They also proved a matching lower bound, which shows that their algorithm achieves the optimal probability of error.

Several recent work [77, 78] improved SE and SR algorithms and the related lower bounds in terms of logarithmic factors or constant factors. In addition, a recent line of research extended the study of this problem to the problem of finding top K best arms or finding the best arms from several disjoint groups of arms. These work also provided algorithms with guarantees in both PAC setting and fixed budget setting [80, 60, 61, 81, 28, 82, 156].

1.3 Matrix completion and tensor completion

A wide range of datasets can be naturally organized in the form of matrices, or, its higher-order generalization, tensors. For example, consider a movie recommendation service, which have collected a large number of users ratings over movies. This dataset can be represented as a matrix where rows correspond to users and columns correspond to movies. Each entry of this matrix is the rating of a movie given by a user. As higher-order generalizations of matrices, tensors can be used to represent data that are addressed by more than two indices. For example, typical movie recommendation services may collect not only rating information, but also side information including genre, time or location. Tensors can be used to represent such datasets, where each entry may represent the rating of a user for a movie on a particular genre (or location).

In many applications, one cannot observe all the entries of the data matrix/tensor. In these applications, it might be too expensive, or simply impossible, to measure all the entries, and some entries might be missing during the measuring process. *Matrix completion* and its generalization *tensor completion* concern the problem of predicting the missing entries in partially observed data matrix or tensor, under certain natural assumptions on the matrix/tensor. The framework of matrix completion and tensor completion capture many real-world applications, including collaborative filtering [65, 107, 141], system identification [100], global positioning [21, 137] and remote sensing [133]. For example, in collaborative filtering, a recommender system may want to predict the missing ratings from a partially observed rating data matrix/tensor, which enables the system to provide personalized recommendations [124, 141, 107]. A well-known example of this application is the Netflix prize, in which the participants were asked to recover a user-movie rating matrix where most of entries are missing [17]. In a localization service, one may want to compute the missing values from a partially observed distance matrix/tensor [138, 96, 21, 137]. In these applications, we observe again the recurring problem of making predictions with a limited number of samples. In the next part, we review several existing algorithms and the related theoretical results of matrix completion and tensor completion.

1.3.1 Matrix completion

Suppose that we only observe a few entries from a matrix. Then, one can easily see that it is impossible to guess the values of missing entries of the matrix without additional assumptions. However, in many instances, the matrix one wish to recover is *low rank* or *approximately low rank*. Indeed, as we have discussed, many data matrices emerged in real-world applications are known to be approximately low rank, e.g., user-by-movie matrices [88, 107], document-by-word matrices [75, 51] and distance matrices [138, 21]. Due the universality of low rankness, the low rank assumption becomes central to the work on matrix completion [32, 31, 35, 119, 83].

Specifically, suppose we wish to recover matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$. Let Ω denote the set of locations corresponding to the observed entries (i.e., $(i, j) \in \Omega$ if and only if M_{ij} is observed). Intuitively, if the number of observations is sufficiently large, we might hope that there is a unique low-rank matrix that agrees with the observations. If this is the case, then one could recover the matrix by solving the following program

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} \quad & \operatorname{rank}(\mathbf{X}), \\ \text{s.t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega. \end{aligned}$$

However, solving this program is an NP-hard problem [44] and therefore it has little practical value. This is also analogous to the NP-hardness of ℓ_0 minimization problems occurred in the literature of compressed sensing [33, 36, 34, 52].

An alternative solution [56, 57, 105, 142] is to use nuclear norm as a surrogate to matrix rank, which leads to the following convex program

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} \quad & \|\mathbf{X}\|_*, \\ \text{s.t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega, \end{aligned} \tag{1.1}$$

where the nuclear norm $\|\mathbf{X}\|_*$ is the sum of all singular values of matrix $\mathbf{X}$. The nuclear norm minimization method was empirically observed to produce very low-rank solutions [56, 55, 15, 105, 141, 142]. Recht et al. [120] showed that Eq. (1.1) almost always produces a rank minimizing solution.

In a seminal work, Candès and Recht [32] showed that, with high probability, by solving Eq. (1.1), one can exactly recover the matrix $\mathbf{M}$ (i.e. $\mathbf{X} = \mathbf{M}$) from $O(n^{1.5} r \log^2(n))$ observations, where $n = \max\{n_1, n_2\}$ and $r = \operatorname{rank}(\mathbf{M})$. Their result assumes that (1) the matrix $\mathbf{M}$ satisfies certain incoherence property (see

Chapter 5) (2) the observations are sampled uniformly at random from all entries $[n_1] \times [n_2]$. Their sample complexity bound was later improved by Candès and Tao [35] and Keshavan et al. [83], who showed that $O(nr\log^2(n))$ observations are sufficient to guarantee recovery. This is a remarkable sample complexity bound since the degree of freedom of a rank r matrix is $O(nr)$, while this result shows that nuclear norm minimization succeeds as soon as the sample size exceeds the degree of freedom by a logarithmic factor.

The proof of Candès and Tao [35] was later simplified by Recht [119], who also slightly improved the sample complexity in terms of constant factors. Candes and Plan [31] considered the noisy matrix completion problem where observations are noisy. They extended the program Eq. (1.1) and obtained a similar sample complexity. Cai et al. [30] proposed the singular value thresholding algorithm which can approximately solve Eq. (1.1) when the matrix $\mathbf{M}$ is very large.

1.3.2 Tensor decomposition and tensor completion

A direct generalization of the matrix completion problem is tensor completion, where the goal is to predict the values of missing entries of a tensor. Due to similar reasons, in order to correctly guess the missing entries, one need certain assumptions on the tensors. A natural assumption is that the tensor to be recovered is low rank.

The definition of tensor rank was first introduced by Hitchcock [74], which is a generalization of the definition of matrix rank as follows. We start from the definition of rank-one tensors. Let $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ be an order-d tensor. Then, tensor $\mathcal{X}$ is rank-one if and only if its entries can be written as

$$X_{i_1,\ldots,i_d} = u_{i_1}^{(1)} u_{i_2}^{(2)} \ldots u_{i_d}^{(d)},$$

where $\mathbf{u}^{(1)} \in \mathbb{R}^{n_1}, \ldots, \mathbf{u}^{(n)} \in \mathbb{R}^{n_d}$ are vectors. Then tensor $\mathcal{T} \in$

$\mathbb{R}^{n_1 \times \ldots \times n_d}$ is rank r, if there exists r rank-one tensors $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(d)}$ such that

$$\mathcal{T} = \mathcal{X}^{(1)} + \ldots + \mathcal{X}^{(d)}.$$

We see that, when $d = 2$, the definition of tensor rank coincides that that of matrix rank.

Empirically, many tensors obtained from real-world applications also have low tensor rank (see [85] and references therein) Naturally, one may consider to generalize the results of matrix completion to the case of tensor completion. However, the properties of tensor rank are known to be fundamentally different from the properties of matrix rank, even for order-3 tensors. In fact, Håstad [71] showed that it is NP-hard to compute the tensor rank. Hillar and Lim [73] further showed that many basic primitives for matrices, for example SVD, cannot be implemented efficiently in the tensor case, even if the tensor is fully observed. The lack of these basic primitives suggests that one cannot directly generalize the results of matrix completion to the tensor case.

One line of research focused on computing approximate low rank decompositions of tensors and obtained many algorithms, including PARAFRAC [70], canonical decomposition [38, 84], tucker decomposition [150] and their extensions [153, 101, 16]. We refer interested readers to [85] for a survey. Many of these algorithms work well empirically. In addition, various heuristics were proposed to apply these low rank models to predict the missing values in a tensor [58, 145]. However, little is known for the theoretical guarantees of these decomposition methods, due to the inherent hardness of tensor rank.

Recently, several work addressed the tensor completion problem by extending the nuclear norm minimization objective Eq. (1.1) developed matrix completion problem [98, 64, 136, 89]. Liu et al. [98] initiated the study of the tensor completion problem by generalizing the definition of nuclear norm to tensors. Several subsequent work authors proposed to optimize a weighted sum of nuclear norms of unfolded matrices of a tensor [64, 136, 146]. However, none of these work guaranteed an exact recovery of tensors and they did not provided a sample complexity bound that is similar to matrix completion. In Chapter 5, we address the tensor completion problem and provide an algorithm with an almost optimal sample complexity. Our approach is based on identifying an alternative low rank model of tensors which is 1) sufficiently rich to capture important information from the datasets and 2) lead to a solvable recovery problem.

1.4 Summary of contributions

In this thesis, we study the learning problems emerging from the application scenarios of multi-armed bandit and tensor completion, where the number of available samples are often limited. Our contributions are summarized as follows.

1.4.1 Combinatorial pure exploration bandits

In Chapter 2, we study the *combinatorial pure exploration (CPE)* problem in the stochastic multi-armed bandit setting, where a learner explores a set of arms with the objective of identifying the optimal member of a *decision class*, which is a collection of subsets of arms with certain combinatorial structures such as size-K subsets, matchings, spanning trees or paths, etc. The CPE problem represents a rich class of pure exploration tasks which covers not only many existing models but also novel cases where the object of interest has a non-trivial combinatorial structure.

We provide a series of results for the general CPE problem. We present general learning algorithms which work for all decision classes that admit offline maximization oracles in both fixed confidence and fixed budget settings. We prove problem-dependent upper bounds of our algorithms. Our analysis exploits the combinatorial structures of the decision classes and introduces a new analytic tool. We also establish a general problem-dependent lower bound for the CPE problem. Our results show that the proposed algorithms achieve the optimal sample complexity (within logarithmic factors) for many decision classes. In addition, applying our results back to the problems of top-K arms identification and multiple bandit best arms identification, we recover the best available upper bounds up to constant factors and partially resolve a conjecture on the lower bounds.

1.4.2 Linear combinatorial bandits

In Chapter 3, we apply the techniques of combinatorial bandits to recommender systems. In particular, we study the cold-start list recommendation problem, in which the objective is to recommend lists of items over a sequence of rounds to a new user with insufficient historical records.

We develop a principled approach called *linear combinatorial bandit* in which a learning algorithm can dynamically identify diverse items that interest a new user. Specifically, each item is represented as a feature vector, and each user is represented as an unknown preference vector. At each of T rounds, the bandit algorithm sequentially selects a set of items according to the item-selection strategy that balances *exploration* and *exploitation*, and collects the user feedback on these selected items. A reward function is further designed to measure the quality (e.g. relevance or diversity) of the selected set based on observed feedback, and the goal of the algorithm is to maximize the total rewards of T rounds. The reward function only needs to satisfy two mild assumptions that is general enough to accommodate a large class of nonlinear functions. To solve this bandit problem, we provide algorithm that achieves $\tilde{O}(\sqrt{T})$ regret after playing T rounds. Experiments conducted on real-wold movie recommendation dataset demonstrate that our approach can effectively address the above challenges and hence improve the performance of recommendation task.

1.4.3 Fast approximation for ridge regression

Ridge regression is a key algorithmic component of our learning algorithm in Chapter 2 and is also central to a wide range of fields including machine learning, data mining and statistics. In practice, an important case is that the number of features p is much larger than the number of samples n, i.e. $p \gg n$. In this case, the popular optimization algorithm for ridge regression runs in $O(n^2p + n^3)$ time, which is expensive when p is very large.

In Chapter 4, we propose a fast *relative-error* approximation algorithm for ridge regression. More specifically, our algorithm outputs a solution $\tilde{\mathbf{x}}$ satisfying $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon\|\mathbf{x}^*\|_2$ with high probability and runs in $\tilde{O}(\mathrm{nnz}(\mathbf{A}) + n^3/\epsilon^2)$ time, where $\mathrm{nnz}(\mathbf{A})$ denotes the number of non-zero elements of matrix $\mathbf{A}$. To the best of our knowledge, this is the first algorithm for ridge regression that runs in $o(n^2p)$ time with provable relative-error approximation bound on the output vector. In addition, for supplements to our main result, we analyze the risk inflation bound of our algorithm and generalize our technique to the multiple response ridge regression problem. Finally, we show empirical results on both synthetic and real datasets.

1.4.4 Recovery for pairwise interaction tensors

In Chapter 5, we study the recovery problem of *pairwise interaction tensors*, a simplified low rank model for tensors. A tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is called a pairwise interaction tensor, if there exists matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{B} \in \mathbb{R}^{n_2 \times n_3}$ and $\mathbf{C} \in \mathbb{R}^{n_3 \times n_1}$ such that each entry $\mathcal{T}$ can be written as the sum of corresponding entries of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, i.e., $T_{ijk} = A_{ij} + B_{jk} + C_{ki}$ for all $i, j, k \in [n_1] \times [n_2] \times [n_3]$.

Due to its simplicity and effectiveness, pairwise interaction tensor has received considerable attention in several applications to be used as a replacement of tensors for modeling multi-attribute dataset. However, in the literature, the existing recovery algorithms for pairwise interaction interaction tensors use local optimization techniques and do not have any guarantee on their recovery performance.

We present an algorithm which, in the absence of noise, guarantees to exactly recover a pairwise interaction tensor from only $O(nr \log^2 n)$ random subsamples of entries, where $n = \max\{n_1, n_2, n_3\}$ and $r = \max\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}), \text{rank}(\mathbf{C})\}$. On the other hand, one can show that the degree of freedom of pairwise interaction tensor is $\Omega(nr)$. Hence, we see that the sample size required by our recovery algorithm is almost minimal.

In addition, for noisy cases, we also present a constrained convex program and prove the associated error bounds. In our analysis, we reformulate the recovery problem as a constrained matrix recovery problem with non-orthonormal operators. Then, we extend the techniques developed by Gross [67] for recovering a matrix from observations of orthonormal operators. Moreover, we develop a simple and scalable approximation optimization algorithm. Our experiments on both synthetic and real datasets agree well with our theory; the experiment results on a real-world recommendation datasets demonstrate state-of-the-art performances.

Chapter 2

Combinatorial Pure Exploration Bandits

We study the *combinatorial pure exploration (CPE)* problem in the stochastic multi-armed bandit setting, where a learner explores a set of arms with the objective of identifying the optimal member of a *decision class*, which is a collection of subsets of arms with certain combinatorial structures such as size-K subsets, matchings, spanning trees or paths, etc. The CPE problem represents a rich class of pure exploration tasks which covers not only many existing models but also novel cases where the object of interest has a non-trivial combinatorial structure. In this chapter, we provide a series of results for the general CPE problem. We present general learning algorithms which work for all decision classes that admit offline maximization oracles in both fixed confidence and fixed budget settings. We prove problem-dependent upper bounds of our algorithms. Our analysis exploits the combinatorial structures of the decision classes and introduces a new analytic tool. We also establish a general problem-dependent lower bound for the CPE problem. Our results show that the proposed algorithms achieve the optimal sample complexity (within logarithmic factors) for many decision classes. In addition, applying our results back to the problems of top-K arms identification and multiple bandit best arms identification, we recover the best available upper bounds up to constant factors and partially resolve a conjecture on the lower bounds.

2.1 Introduction

Multi-armed bandit (MAB) is a predominant model for characterizing the tradeoff between exploration and exploitation in decision-making problems. Although this is an intrinsic tradeoff in many tasks, some application domains prefer a dedicated exploration procedure in which the goal is to identify an optimal object among a collection of candidates and the reward or loss incurred during exploration is irrelevant. In light of these applications, the related learning problem, called pure exploration in MABs, has received much attention. Recent advances in pure exploration MABs have found potential applications in many domains including crowdsourcing, communication network and online advertising.

In many of these application domains, a recurring problem is to identify the optimal object with certain *combinatorial structure*. For example, a crowdsourcing application may want to find the best assignment from workers to tasks such that overall productivity of workers are maximized. A network routing system during the initialization phase may try to build a spanning tree that minimizes the delay of links, or attempts to identify the shortest path between two sites. An online advertising system may be interested in finding the best matching between ads and display slots. The literature of pure exploration MAB problems lacks a framework that encompasses these kinds of problems where the object of interest has a non-trivial combinatorial structure. This chapter contributes such a framework which accounts for general combinatorial structures, and develops a series of results, including algorithms, upper bounds and lower bounds for the framework.

In this chapter, we formulate the *combinatorial pure exploration (CPE)* problem for stochastic multi-armed bandits. In the CPE problem, a learner has a fixed set of arms and each arm is associated with an unknown reward distribution. The learner is also given a collection of sets of arms called *decision class*, which corresponds to a collection of certain combinatorial structures. During the exploration period, in each round the learner chooses an arm to play and observes a random reward sampled from the associated distribution. The objective is when the exploration period ends, the learner outputs a member of the decision class that she believes to be optimal, in the sense that the sum of expected rewards of all arms in the output set is maximized among all members in the decision class.

The CPE framework represents a rich class of pure exploration problems. The conventional pure exploration problem in MAB, whose objective is to find the single best arm, clearly fits into this framework, in which the decision class is the collection of all singletons. This framework also naturally encompasses several recent extensions, including the problem of finding the top K arms (henceforth TopK) [80, 81, 28, 82, 156] and the multi-bandit problem of finding the best arms simultaneously from several disjoint sets of arms (henceforth MB) [60, 28]. Further, this framework covers many more interesting cases where the decision classes correspond to collections of non-trivial combinatorial structures. For example, suppose that the arms represent the edges in a graph. Then a decision class could be the set of all paths between two vertices, all spanning trees or all matchings of the graph. And, in these cases, the objectives of CPE become identifying the optimal paths, spanning trees and matchings through bandit explorations, respectively. To our knowledge, there are no results available in the literature for these pure exploration tasks.

The CPE framework raises several interesting challenges to the design and analysis of pure exploration algorithms. One challenge is that, instead of solving each type of CPE task in an ad-hoc way, one requires a unified algorithm and analysis that support different decision classes. Another challenge stems from the combinatorial nature of CPE, namely that the optimal set may contain some arms with very small expected rewards (e.g., it is possible that a maximum matching contains the edge with the smallest weight); hence, arms cannot be eliminated simply based on their own rewards in the learning algorithm or ignored in the analysis. This differs from many existing approach of pure exploration MABs. Therefore, the design and analysis of algorithms for CPE demands novel techniques which take both rewards and combinatorial structures into account.

Our results. In this chapter, we propose two novel learning algorithms for general CPE problem: one for the fixed confidence setting and one for the fixed budget setting. Both algorithms support a wide range of decision classes in a unified way. In the fixed confidence setting, we present Combinatorial Lower-Upper Confidence Bound (CLUCB) algorithm. The CLUCB algorithm does not need to know the definition of the decision class, as long as it has access to the decision class through a maximization oracle. We upper bound the number of samples used by CLUCB. This sample complexity bound depends on both the expected rewards and the structure of decision class. Our analysis relies on a novel combinatorial construction called *exchange class*, which may be of independent interest for other combinatorial optimization problems. Specializing our result to TopK and MB, we recover the best available sample complexity bounds [81, 61, 82] up to constant factors. While for other decision classes in general, our result establishes the first sample complexity upper bound. We further show that CLUCB can be easily extended to the fixed budget setting and PAC learning setting and we provide related theoretical guarantees in the Section 2.9.

Moreover, we establish a problem-dependent sample complexity lower bound for the CPE problem. Our lower bound shows that the sample complexity of the proposed CLUCB algorithm is optimal (to within logarithmic factors) for many decision classes, including TopK, MB and the decision classes derived from matroids (e.g., spanning tree). Therefore our upper and lower bounds provide a nearly full characterization of the sample complexity of these CPE problems. For more general decision classes, our results show that the upper and lower bounds are within a relatively benign factor. To the best of our knowledge, there are no problem-dependent lower bounds known for pure exploration MABs besides the case of identifying the single best arm [102, 6]. We also notice that our result resolves the conjecture of Bubeck et al. [28] on the problem-dependent sample complexity lower bounds of TopK and MB problems, for the cases of Gaussian reward distributions.

In the fixed budget setting, we present a parameter-free algorithm called Combinatorial Successive Accept Reject (CSAR) algorithm. We prove a probability of error bound of the CSAR algorithm. This bound can be shown to be equivalent to the sample complexity bound of CLUCB within logarithmic factors, although the two algorithms are based on quite different techniques. Our analysis of CSAR re-uses exchange classes as tools. This suggests that exchange classes may be useful for analyzing similar problems. In addition, when applying the algorithm to back TopK and MB, our bound recovers the best known result in the fixed budget setting due to Bubeck et al. [28] up to constant factors.

2.2 Problem Formulation

In this section, we formally define the CPE problem. Suppose that there are n arms and the arms are numbered $1, 2, \ldots, n$. Assume that each arm $e \in [n]$ is associated with a reward distribution φ_e. Let $\boldsymbol{w} = \big(w(1), \ldots, w(n)\big)^T$ denote the vector of expected rewards, where each entry $w(e) = \mathbb{E}_{X\sim\varphi_e}[X]$ denotes the expected reward of arm e. Following standard assumptions of stochastic MABs, we assume that all reward distributions have R-sub-Gaussian tails for some known constant $R > 0$. Formally, if X is a random variable drawn from φ_e for some $e \in [n]$, then, for all $t \in \mathbb{R}$, one has $\mathbb{E}\big[\exp(tX - t\mathbb{E}[X])\big] \leq \exp(R^2t^2/2)$. It is known that the family of R-sub-Gaussian tail distributions encompasses all distributions that are supported on $[0, R]$ as well as many unbounded distributions such as Gaussian distributions with variance R^2 (see e.g., [114, 125]).

We define a *decision class* $\mathcal{M} \subseteq 2^{[n]}$ as a collection of sets of arms. Let $M_* = \arg\max_{M\in\mathcal{M}} w(M)$ denote the optimal member of the decision class $\mathcal{M}$ which maximizes the sum of expected rewards[1]. A learner's objective is to identify M_* from $\mathcal{M}$ by playing the following game with the stochastic environment. At the beginning of the game, the decision class $\mathcal{M}$ is revealed to the learner while the reward distributions $\{\varphi_e\}_{e\in[n]}$ are unknown to the learner. Then, the learner plays the game over a sequence of rounds; in each round t, the learner pulls an arm $p_t \in [n]$ and observes a reward sampled from the associated reward distribution φ_{p_t}. The game continues until certain stopping condition is satisfied. After the game finishes, the learner need to output a set $\mathsf{Out} \in \mathcal{M}$.

We consider two different stopping conditions of the game, which are known as *fixed confidence* setting and *fixed budget* set-

[1] We define $v(S) \triangleq \sum_{i\in S} v(i)$ for any vector $\boldsymbol{v} \in \mathbb{R}^n$ and any set $S \subseteq [n]$. In addition, for convenience, we will assume that M_* is unique.

ting in the literature. In the fixed confidence setting, the learner can stop the game at any round. The learner need to guarantee that $\Pr[\mathsf{Out} = M_*] \geq 1 - \delta$ for a given confidence parameter δ. The learner's performance is evaluated by her *sample complexity*, i.e., the number of pulls used by the learner. In the fixed budget setting, the game stops after a fixed number T of rounds, where T is given before the game starts. The learner tries to minimize the *probability of error*, which is formally $\Pr[\mathsf{Out} \neq M_*]$, within T rounds. In this case, the learner's performance is measured by the probability of error.

2.3 Algorithm, Exchange Class and Sample Complexity

In this section, we present Combinatorial Lower-Upper Confidence Bound (CLUCB) algorithm, a learning algorithm for the CPE problem in the fixed confidence setting, and analyze its sample complexity. En route to our sample complexity bound, we introduce the notions of exchange classes and the widths of decision classes, which play an important role in the analysis and sample complexity bound. Furthermore, the CLUCB algorithm can be extended to the fixed budget and PAC learning settings, the discussion of which is included in Section 2.9.

Oracle. We allow the CLUCB algorithm to access a *maximization oracle*. A maximization oracle takes a weight vector $\boldsymbol{v} \in \mathbb{R}^n$ as input and finds an optimal set from a given decision class $\mathcal{M}$ with respect to the weight vector $\boldsymbol{v}$. Formally, we call a function Oracle: $\mathbb{R}^n \to \mathcal{M}$ a maximization oracle for $\mathcal{M}$ if, for all $\boldsymbol{v} \in \mathbb{R}^n$, we have $\mathrm{Oracle}(\boldsymbol{v}) \in \arg\max_{M \in \mathcal{M}} v(M)$. It is clear that a wide range of decision classes admit such maximization oracles, including decision classes corresponding to collections of matchings, paths or bases of matroids (see later for concrete examples). Besides the access to the oracle, CLUCB does not need *any* additional knowledge of the decision class $\mathcal{M}$.

Algorithm. Now we describe the details of CLUCB, as shown in Algorithm 1. During its execution, the CLUCB algorithm maintains empirical mean $\bar{w}_t(e)$ and confidence radius $\mathrm{rad}_t(e)$ for each arm $e \in [n]$ and each round t. The construction of confidence radius ensures that $|w(e) - \bar{w}_t(e)| \leq \mathrm{rad}_t(e)$ holds with high probability for each arm $e \in [n]$ and each round $t > 0$. CLUCB begins with an initialization phase in which each arm is pulled once. Then, at round $t \geq n$, CLUCB uses the following procedure to choose an arm to play. First, CLUCB calls the oracle which finds the set $M_t = \mathrm{Oracle}(\bar{\boldsymbol{w}}_t)$. The set M_t is the "best" set with respect to the empirical means $\bar{\boldsymbol{w}}_t$. Then, CLUCB explores possible refinements of M_t. In particular, CLUCB uses the confidence radius to compute an adjusted expectation vector $\tilde{\boldsymbol{w}}_t$ in the following way: for each arm $e \in M_t$, $\tilde{w}_t(e)$ is equal to to the lower confidence bound $\tilde{w}_t(e) = \bar{w}_t(e) - \mathrm{rad}_t(e)$; and for each arm $e \notin M_t$, $\tilde{w}_t(e)$ is equal to the upper confidence bound $\tilde{w}_t(e) = \bar{w}_t(e) + \mathrm{rad}_t(e)$. Intuitively, the adjusted expectation vector $\tilde{\boldsymbol{w}}_t$ penalizes arms belonging to the current set M_t and encourages exploring arms out of M_t. CLUCB then calls the oracle using the adjusted expectation vector $\tilde{\boldsymbol{w}}_t$ as input to compute a refined set $\tilde{M}_t = \mathrm{Oracle}(\tilde{\boldsymbol{w}}_t)$. If $\tilde{w}_t(\tilde{M}_t) = \tilde{w}_t(M_t)$ then CLUCB stops and returns $\mathsf{Out} = M_t$. Otherwise, CLUCB pulls the arm that belongs to the symmetric difference between M_t and $\tilde{M}_t$ and has the largest confidence radius (intuitively the largest uncertainty). This ends the t-th round of CLUCB. We note that CLUCB generalizes and unifies the ideas of several different fixed confidence algorithms dedicated to the TopK and MB problems in the literature [81, 61, 82].

Algorithm 1 CLUCB: Combinatorial Lower-Upper Confidence Bound

```
Require: Confidence $\delta \in (0,1)$; Maximization oracle: Oracle$(\cdot) : \mathbb{R}^n \to \mathcal{M}$
    Initialize: Play each arm $e \in [n]$ once. Initialize empirical means $\bar{\boldsymbol{w}}_n$ and set $T_n(e) \leftarrow 1$ for all $e$.
for $t = n, n+1, \ldots$ do
    $M_t \leftarrow$ Oracle$(\bar{\boldsymbol{w}}_t)$
    Compute confidence radius $\mathrm{rad}_t(e)$ for all $e \in [n]$   ▷ $\mathrm{rad}_t(e)$ is defined later in Theorem 2.1
    for $e = 1, \ldots, n$ do
        if $e \in M_t$ then $\tilde{w}_t(e) \leftarrow \bar{w}_t(e) - \mathrm{rad}_t(e)$
        else $\tilde{w}_t(e) \leftarrow \bar{w}_t(e) + \mathrm{rad}_t(e)$
    end for
    $\tilde{M}_t \leftarrow$ Oracle$(\tilde{\boldsymbol{w}}_t)$
    if $\tilde{w}_t(\tilde{M}_t) = \tilde{w}_t(M_t)$ then
        Out $\leftarrow M_t$
        return Out
    end if
    $p_t \leftarrow \arg\max_{e \in (\tilde{M}_t \setminus M_t) \cup (M_t \setminus \tilde{M}_t)} \mathrm{rad}_t(e)$   ▷ break ties arbitrarily
    Pull arm $p_t$ and observe the reward
    Update empirical means $\bar{\boldsymbol{w}}_{t+1}$ using the observed reward
    Update number of pulls: $T_{t+1}(p_t) \leftarrow T_t(p_t) + 1$ and $T_{t+1}(e) \leftarrow T_t(e)$ for all $e \neq p_t$
end for
```

2.3.1 Sample complexity

Now we establish a problem-dependent sample complexity bound of the CLUCB algorithm. To formally state our result, we need to introduce several notions.

Gap. We begin with defining a natural hardness measure of the CPE problem. For each arm $e \in [n]$, we define its gap Δ_e as

$$\Delta_e = \begin{cases} w(M_*) - \max_{M \in \mathcal{M}: e \in M} w(M) & \text{if } e \notin M_*, \\ w(M_*) - \max_{M \in \mathcal{M}: e \notin M} w(M) & \text{if } e \in M_*, \end{cases} \tag{2.1}$$

where we adopt the convention that the maximum value of an empty set is $-\infty$. We also define the hardness $\mathbf{H}$ as the sum of inverse squared gaps

$$\mathbf{H} = \sum_{e \in [n]} \Delta_e^{-2}. \tag{2.2}$$

From Eq. (2.1), we see that, for each arm $e \notin M_*$, the gap Δ_e represents the sub-optimality of the best set that includes arm e; and, for each arm $e \in M_*$, the gap Δ_e is the sub-optimality of the best set that does not include arm e. This naturally generalizes and unifies previous definitions of gaps [6, 60, 80, 28].

Exchange class and the width of a decision class. A notable challenge of our analysis stems from the generality of CLUCB which, as we have seen, supports a wide range of decision classes $\mathcal{M}$. Indeed, previous algorithms for special cases including TopK and MB require a separate analysis for each individual type of problem. Such strategy is intractable for our setting and we need a unified analysis for all decision classes. Our solution to this challenge is a novel combinatorial construction called *exchange class*, which is used as a proxy for the structure of the decision class. Intuitively, an exchange class $\mathcal{B}$ for a decision class $\mathcal{M}$ can be seen as a collection of "patches" (borrowing concepts from source code management) such that, for any two different sets $M, M' \in \mathcal{M}$, one can transform M to M' by applying a series of patches of $\mathcal{B}$; and each application of a patch yields a valid member of $\mathcal{M}$. These patches are later used by our analysis to build gadgets that interpolate between different members of the decision class and serve to bridge key quantities. Furthermore, the maximum patch size of $\mathcal{B}$ will play an important role in our sample complexity bound.

Now we formally define the exchange class. We begin with the definition of exchange sets, which formalize the aforementioned "patches". We define an exchange set b as an ordered pair of disjoint sets $b = (b_+, b_-)$ where $b_+ \cap b_- = \emptyset$ and $b_+, b_- \subseteq [n]$. Then, we define operator $\oplus$ such that, for any set $M \subseteq [n]$ and any exchange set $b = (b_+, b_-)$, we have $M \oplus b \triangleq M \backslash b_- \cup b_+$. Similarly, we also define operator $\ominus$ such that $M \ominus b \triangleq M \backslash b_+ \cup b_-$.

We call a collection of exchange sets $\mathcal{B}$ an *exchange class for* $\mathcal{M}$ if $\mathcal{B}$ satisfies the following property. For any $M, M' \in \mathcal{M}$ such that $M \neq M'$ and for any $e \in (M \backslash M')$, there exists an exchange set $(b_+, b_-) \in \mathcal{B}$ which satisfies five constraints: **(a)** $e \in b_-$, **(b)** $b_+ \subseteq M' \backslash M$, **(c)** $b_- \subseteq M \backslash M'$, **(d)** $(M \oplus b) \in \mathcal{M}$ and **(e)** $(M' \ominus b) \in \mathcal{M}$.

Intuitively, constraints **(b)** and **(c)** resemble the concept of patches in the sense that b_+ contains only the "new" elements from M' and b_- contains only the "old" elements of M; constraints **(d)** and **(e)** allow one to transform M one step closer to M' by applying a patch $b \in \mathcal{B}$ to yield $(M \oplus b) \in \mathcal{M}$ (and similarly for $M' \ominus b$). These transformations are the basic building blocks in our analysis. Furthermore, as we will see later in our examples, for many decision classes, there are exchange classes representing natural combinatorial structures, e.g., augmenting paths and cycles of matchings.

In our analysis, the key quantity of exchange class is called *width*, which is defined as the size of the largest exchange set as follows

$$\mathrm{width}(\mathcal{B}) = \max_{(b_+, b_-) \in \mathcal{B}} |b_+| + |b_-|. \tag{2.3}$$

Let $\mathrm{Exchange}(\mathcal{M})$ denote the family of all possible exchange classes for $\mathcal{M}$. We define the width of a decision class $\mathcal{M}$ as the width of the thinnest exchange class

$$\mathrm{width}(\mathcal{M}) = \min_{\mathcal{B} \in \mathrm{Exchange}(\mathcal{M})} \mathrm{width}(\mathcal{B}). \tag{2.4}$$

Sample complexity. Our main result of this section is a problem-dependent sample complexity bound of the CLUCB algorithm which show that, with high probability, CLUCB returns the optimal set M_* and uses at most $\tilde{O}\big(\mathrm{width}(\mathcal{M})^2 \mathbf{H}\big)$ samples.

Theorem 2.1. *Given any $\delta \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\boldsymbol{w} \in \mathbb{R}^n$. Assume that the reward distribution φ_e for each arm $e \in [n]$ has mean $w(e)$ with an R-sub-Gaussian tail. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ denote the optimal set. Set $\mathrm{rad}_t(e) = R\sqrt{2\log\left(\frac{4nt^3}{\delta}\right)/T_t(e)}$ for all $t > 0$ and $e \in [n]$. Then, with probability at least $1-\delta$, the* CLUCB *algorithm (Algorithm 1) returns the optimal set* $\mathsf{Out} = M_*$ *and*

$$T \le O\left(R^2 \mathit{width}(\mathcal{M})^2 \mathbf{H} \log\left(R^2\mathbf{H}/\delta\right)\right), \tag{2.5}$$

where T denotes the number of samples used by Algorithm 1, $\mathbf{H}$ is defined in Eq. (2.2) *and $\mathit{width}(\mathcal{M})$ is defined in Eq.* (2.4)*.*

The proof of Theorem 2.1 is deferred to Section 2.8.

2.3.2 Examples of decision classes

Now we investigate several concrete types of decision classes, which correspond to different CPE tasks. We analyze the width of these decision classes and apply Theorem 2.1 to obtain the sample complexity bounds. A more detailed analysis and the constructions of exchange classes is deferred to Section 2.13. We begin with the problem of top-K arm identification (TopK) and multi-bandit best arms identification (MB).

Example 2.1 (TopK and MB). *For any $K \in [n]$, the problem of finding the top K arms with the largest expected reward can be modeled by decision class $\mathcal{M}_{\mathrm{TopK}(K)} = \{M \subseteq [n] \mid |M| = K\}$. Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a partition of $[n]$. The problem of identifying the best arms from each group of arms $A_1, \ldots, A_m$ can be modeled by decision class $\mathcal{M}_{\mathrm{MB}(\mathcal{A})} = \{M \subseteq [n] \mid \forall i \in [m], |M \cap A_i| = 1\}$. Note that maximization oracles for these two decision classes are trivially the functions of returning the best arms or the best arm of each group.*

Then we have $\mathit{width}(\mathcal{M}_{\mathrm{TopK}(K)}) \le 2$ and $\mathit{width}(\mathcal{M}_{\mathrm{MB}(\mathcal{A})}) \le 2$ (see Fact 2.2 and 2.3 in Section 2.3.2) and therefore the sample *complexity of* CLUCB *for solving* TopK *and* MB *is* $O\big(\mathbf{H}\log(\mathbf{H}/\delta)\big)$*, which matches previous results in the fixed confidence setting [81, 61, 82] up to constant factors.*

Next we consider the problem of identifying the maximum matching or the problem of finding the shortest path (by negating the rewards) in a setting where arms correspond to edges. For these problems, Theorem 2.1 establishes the first known sample complexity bound.

Example 2.2 (Matchings and Paths)**.** *Let* $G(V, E)$ *be a graph with* n *edges and assume there is a one-to-one mapping between edges* E *and arms* $[n]$*. Suppose that* G *is a bipartite graph. Let* $\mathcal{M}_{\text{MATCH}(G)}$ *correspond to the set of all matchings in* G*. Then we have width*$(\mathcal{M}_{\text{MATCH}(G)}) \le |V|$ *(In fact, we construct an exchange class corresponding to the collection of augmenting cycles and augmenting paths of* G*; see Fact 2.4).*

Next suppose that G *is a directed acyclic graph and let* $s, t \in V$ *be two vertices. Let* $\mathcal{M}_{\text{PATH}(G,s,t)}$ *correspond to the set of all paths from* s *to* t*. Then we have width*$(\mathcal{M}_{\text{PATH}(G,s,t)}) \le |V|$ *(In fact, we construct an exchange class corresponding to the collection of disjoint pairs of paths; see Fact 2.5). Therefore the sample complexity bounds of* CLUCB *for decision classes* $\mathcal{M}_{\text{MATCH}(G)}$ *and* $\mathcal{M}_{\text{PATH}(G,s,t)}$ *are* $O\big(|V|^2\mathbf{H}\log(\mathbf{H}/\delta)\big)$*.*

Last, we investigate the general problem of identifying the maximum-weight basis of a matroid. Again, Theorem 2.1 is the first sample complexity upper bound for this type of pure exploration tasks.

Example 2.3 (Matroids)**.** *Let* $T = (E, \mathcal{I})$ *be a finite matroid, where* E *is a set of size* n *(called ground set) and* $\mathcal{I}$ *is a family of subsets of* E *(called independent sets) which satisfies the axioms of matroids. Assume that there is a one-to-one mapping between* E *and* $[n]$*. Recall that a basis of matroid* T *is a maximal independent set. Let* $\mathcal{M}_{\text{MATROID}(T)}$ *correspond to the set of all* *bases of T. Then we have width$(\mathcal{M}_{\textsc{Matroid}(T)}) \leq 2$ (derived from strong basis exchange property of matroids; see Fact 2.1) and the sample complexity of* CLUCB *for $\mathcal{M}_{\textsc{Matroid}(T)}$ is $O\big(\mathbf{H}\log(\mathbf{H}/\delta)\big)$.*

The last example $\mathcal{M}_{\textsc{Matroid}(T)}$ is a general type of decision class which encompasses many pure exploration tasks including TopK and MB as special cases, where TopK corresponds to uniform matroids of rank K and MB corresponds to partition matroids. It is easy to see that $\mathcal{M}_{\textsc{Matroid}(T)}$ also covers the decision class that contains all spanning trees of a graph. On the other hand, it has been established that matchings and paths cannot be formulated as matroids since they are matroid intersections [112].

2.4 Lower Bound

In this section, we present a problem-dependent lower bound on the sample complexity of the CPE problem. To state our results, we first define the notion of *δ-correct algorithm* as follows. For any $\delta \in (0,1)$, we call an algorithm $\mathbb{A}$ a δ-correct algorithm if, for any expected reward $\boldsymbol{w} \in \mathbb{R}^n$, the probability of error of $\mathbb{A}$ is at most δ, i.e., $\Pr[M_* \neq \mathsf{Out}] \leq \delta$, where Out is the output of $\mathbb{A}$.

We show that, for any decision class $\mathcal{M}$ and any expected rewards $\boldsymbol{w}$, a δ-correct algorithm $\mathbb{A}$ must use at least $\Omega\big(\mathbf{H}\log(1/\delta)\big)$ samples in expectation.

Theorem 2.2. *Fix any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any vector $\boldsymbol{w} \in \mathbb{R}^n$. Suppose that, for each arm $e \in [n]$, the reward distribution φ_e is given by $\varphi_e = \mathcal{N}(w(e), 1)$, where we let $\mathcal{N}(\mu, \sigma^2)$ denote Gaussian distribution with mean μ and variance σ^2. Then, for any $\delta \in (0, e^{-16}/4)$ and any δ-correct algorithm $\mathbb{A}$, we have*

$$\mathbb{E}[T] \geq \frac{1}{16}\mathbf{H}\log\left(\frac{1}{4\delta}\right), \tag{2.6}$$

where T denote the number of total samples used by algorithm $\mathbb{A}$ and $\mathbf{H}$ is defined in Eq. (2.2)*.*

In Example 2.1 and Example 2.3, we have seen that the sample complexity of CLUCB is $O(\mathbf{H}\log(n\mathbf{H}/\delta))$ for pure exploration tasks including TopK, MB and more generally the CPE tasks with decision classes derived from matroids, i.e., $\mathcal{M}_{\text{Matroid}(T)}$ (including spanning trees). Hence, our upper and lower bound show that the CLUCB algorithm achieves the optimal sample complexity within logarithmic factors for these pure exploration tasks. In addition, we remark that Theorem 2.2 resolves the conjecture of Bubeck et al. [28] that the lower bounds of sample complexity of TopK and MB problems are $\Omega\big(\mathbf{H}\log(1/\delta)\big)$, for the cases of Gaussian reward distributions.

On the other hand, for general decision classes with non-constant widths, we see that there is a gap of $\tilde{\Theta}(\text{width}(\mathcal{M})^2)$ between the upper bound Eq. (2.5) and the lower bound Eq. (2.6). Notice that we have $\text{width}(\mathcal{M}) \leq n$ for any decision class $\mathcal{M}$ and therefore the gap is relatively benign. Our lower bound also suggests that the dependency on $\mathbf{H}$ of the sample complexity of CLUCB cannot be improved up to logarithmic factors. Furthermore, we conjecture that the sample complexity lower bound might inherently depend on the size of exchange sets. In Section 2.10.2, we provide evidences on this conjecture which is a lower bound on the sample complexity of exploration of the exchange sets.

2.5 Fixed Budget Algorithm

In this section, we present Combinatorial Successive Accept Reject (CSAR) algorithm, which is a parameter-free learning algorithm for the CPE problem in the fixed budget setting. Then, we upper bound the probability of error CSAR in terms of gaps and $\text{width}(\mathcal{M})$.

Constrained oracle. The CSAR algorithm requires access to a *constrained oracle*, which is a function denoted as COracle : $\mathbb{R}^n \times 2^{[n]} \times 2^{[n]} \to \mathcal{M} \cup \{\perp\}$ and satisfies

$$\mathrm{COracle}(\boldsymbol{v}, A, B) = \begin{cases} \arg\max_{M \in \mathcal{M}_{A,B}} v(M) & \text{if } \mathcal{M}_{A,B} \neq \emptyset \\ \perp & \text{if } \mathcal{M}_{A,B} = \emptyset, \end{cases} \tag{2.7}$$

where we define $\mathcal{M}_{A,B} = \{M \in \mathcal{M} \mid A \subseteq M, B \cap M = \emptyset\}$ as the collection of feasible sets and $\perp$ is a null symbol. Hence we see that COracle$(\boldsymbol{v}, A, B)$ returns an optimal set that includes all elements of A while excluding all elements of B; and if there are no feasible sets, the constrained oracle COracle$(\boldsymbol{v}, A, B)$ returns the null symbol $\perp$. In Section 2.14, we show that constrained oracles are equivalent to maximization oracles up to a transformation on the weight vector. In addition, similar to CLUCB, CSAR does not need any additional knowledge of $\mathcal{M}$ other than accesses to a constrained oracle for $\mathcal{M}$.

Algorithm. The idea of the CSAR algorithm is as follows. The CSAR algorithm divides the budget of T rounds into n phases. In the end of each phase, CSAR either accepts or rejects a single arm. If an arm is accepted, then it is included into the final output. Conversely, if an arm is rejected, then it is excluded from the final output. The arms that are neither accepted nor rejected are sampled for an equal number of times in the next phase.

Now we describe the procedure of the CSAR algorithm for choosing an arm to accept/reject. Let A_t denote the set of accepted arms before phase t and let B_t denote the set of rejected arms before phase t. We call an arm e to be active if $e \notin A_t \cup B_t$. In the beginning of phase t, CSAR samples each active arm for $\tilde{T}_t - \tilde{T}_{t-1}$ times, where the definition of $\tilde{T}_t$ is given in Algorithm 2. Next, CSAR calls the constrained oracle to compute an optimal set M_t with respect to the empirical means $\bar{\boldsymbol{w}}_t$, accepted arms A_t and rejected arms B_t, i.e., $M_t = \mathrm{COracle}(\bar{\boldsymbol{w}}_t, A_t, B_t)$. It is clear that the output of COracle$(\bar{\boldsymbol{w}}_t, A_t, B_t)$ is independent from the input $\bar{w}_t(e)$ for any $e \in A_t \cup B_t$. Then, for each active arm e, CSAR estimates the "empirical gap" of e in the following way. If $e \in M_t$, then CSAR computes an optimal set $\tilde{M}_{t,e}$ that does not include e, i.e., $\tilde{M}_{t,e} =$ COracle$(\bar{\boldsymbol{w}}_t, A_t, B_t \cup \{e\})$. Conversely, if $e \notin M_t$, then CSAR computes an optimal $\tilde{M}_{t,e}$ which includes e, i.e., $\tilde{M}_{t,e} =$ COracle$(\bar{\boldsymbol{w}}_t, A_t \cup \{e\}, B_t)$. Then, the empirical gap of e is calculated as $\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e})$. Finally, CSAR chooses the arm p_t which has the largest empirical gap. If $p_t \in M_t$ then p_t is accepted, otherwise p_t is rejected. The pseudo-code CSAR is shown in Algorithm 2. We note that CSAR can be considered as a generalization of the ideas of the two versions of SAR algorithm due to Bubeck et al. [28], which are designed specifically for the TopK and MB problems respectively.

2.5.1 Probability of error

In the following theorem, we bound the probability of error of the CSAR algorithm.

Theorem 2.3. *Given any $T > n$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\boldsymbol{w} \in \mathbb{R}^n$. Assume that the reward distribution φ_e for each arm $e \in [n]$ has mean $w(e)$ with an R-sub-Gaussian tail. Let $\Delta_{(1)}, \ldots, \Delta_{(n)}$ be a permutation of $\Delta_1, \ldots, \Delta_n$ (defined in Eq. (2.1)) such that $\Delta_{(1)} \leq \ldots \ldots \Delta_{(n)}$. Define $\mathbf{H}_2 \triangleq \max_{i \in [n]} i \Delta_{(i)}^{-2}$. Then, the CSAR algorithm uses at most T samples and outputs a solution $\mathsf{Out} \in \mathcal{M} \cup \{\perp\}$ such that*

$$\Pr[\mathsf{Out} \neq M_*] \leq n^2 \exp\left(-\frac{(T-n)}{18R^2 \tilde{\log}(n)\, width(\mathcal{M})^2 \mathbf{H}_2}\right), \quad (2.8)$$

where $\tilde{\log}(n) \triangleq \sum_{i=1}^{n} i^{-1}$, $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ and $width(\mathcal{M})$ is defined in Eq. (2.4).

The proof of Theorem 2.3 is deferred to Section 2.11.

Algorithm 2 CSAR: Combinatorial Successive Accept Reject

```
Require: Budget: $T > 0$; Constrained oracle: COracle : $\mathbb{R}^n \times 2^{[n]} \times 2^{[n]} \to \mathcal{M} \cup \{\perp\}$.
Define $\tilde{\log}(n) \triangleq \sum_{i=1}^{n} \frac{1}{i}$
$\tilde{T}_0 \leftarrow 0, A_1 \leftarrow \emptyset, B_1 \leftarrow \emptyset$
for $t = 1, \ldots, n$ do
    $\tilde{T}_t \leftarrow \left\lceil \frac{T-n}{\tilde{\log}(n)(n-t+1)} \right\rceil$
    Pull each arm $e \in [n] \backslash (A_t \cup B_t)$ for $\tilde{T}_t - \tilde{T}_{t-1}$ times
    Update the empirical means $\bar{\boldsymbol{w}}_t$ for each arm $e \in [n] \backslash (A_t \cup B_t)$   ▷ set $\bar{w}_t(e) = 0,\ \forall e \in A_t \cup B_t$
    $M_t \leftarrow$ COracle($\bar{\boldsymbol{w}}_t, A_t, B_t$)
    if $M_t = \perp$ then
        fail: set Out $\leftarrow \perp$ and return Out
    end if
    for each $e \in [n] \backslash (A_t \cup B_t)$ do
        if $e \in M_t$ then $\tilde{M}_{t,e} \leftarrow$ COracle($\bar{\boldsymbol{w}}_t, A_t, B_t \cup \{e\}$)
        else $\tilde{M}_{t,e} \leftarrow$ COracle($\bar{\boldsymbol{w}}_t, A_t \cup \{e\}, B_t$)
    end for
    $p_t \leftarrow \arg\max_{e \in [n] \backslash (A_t \cup B_t)} \bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e})$   ▷ define $\bar{w}_t(\perp) = -\infty$; break ties arbitrarily
    if $p_t \in M_t$ then
        $A_{t+1} \leftarrow A_t \cup \{p_t\}, B_{t+1} \leftarrow B_t$
    else
        $A_{t+1} \leftarrow A_t, B_{t+1} \leftarrow B_t \cup \{p_t\}$
    end if
end for
Out $\leftarrow A_{n+1}$
return Out
```

One can verify that $\mathbf{H}_2$ is equivalent to $\mathbf{H}$ up to a logarithmic factor: $\mathbf{H}_2 \leq \mathbf{H} \leq \log(2n)\mathbf{H}_2$ (see [6]). Therefore, by setting the probability of error (the RHS of Eq. (2.8)) to a constant, one can see that CSAR requires a budget of $T = \tilde{O}(\mathrm{width}(\mathcal{M})^2\mathbf{H})$ samples. This is equivalent to the sample complexity bound of CLUCB up to logarithmic factors. In addition, applying Theorem 2.3 back to TopK and MB, our bound matches the previous fixed budget algorithm due to Bubeck et al. [28].

2.6 Related Work

The multi-armed bandit problem has been extensively studied in both stochastic and adversarial settings [91, 10, 9]. We refer readers to [25] for a survey on recent advances. Many work in MABs focus on minimizing the cumulative regret, which is an objective known to be fundamentally different from the objective of pure exploration MABs [26]. Among these work, a recent line of research considers a generalized setting called combinatorial bandits in which a set of arms (satisfying certain combinatorial constraints) are played on each round [39, 79, 111, 27, 43, 66, 95, 90]. Note that the objective of these work is to minimize the cumulative regret, which differs from ours.

In the literature of pure exploration MABs, the classical problem of identifying the single best arm has been well-studied in both fixed confidence and fixed budget settings [102, 53, 26, 6, 61, 77, 78]. A flurry of recent work extend this classical problem to TopK and MB problems and obtain algorithms with upper bounds [80, 60, 61, 81, 28, 82, 156] and worst-case lower bounds of TopK [81, 156]. Our framework encompasses these two problems as special cases and covers a much larger class of combinatorial pure exploration problems, which have not been addressed in current literature. Applying our results back to TopK and MB, our upper bounds match best available problem-dependent bounds up to constant factors [61, 81, 28] in both fixed confidence and fixed budget settings; and our lower bound is the first proven problem-dependent lower bound for these two problems, which are conjectured earlier by Bubeck et al. [28].

2.7 Conclusion

In this chapter, we proposed a general framework called combinatorial pure exploration (CPE) that can handle pure exploration tasks for many complex bandit problems with combinatorial constraints, and have potential applications in various domains. We have shown a number of results for the framework, including two novel learning algorithms, their related upper bounds and a novel lower bound. The proposed algorithms support a wide range of decision classes in a unifying way and our analysis introduced a novel tool called exchange class, which may be of independent interest. Our upper and lower bounds characterize the complexity of the CPE problem: the sample complexity of our algorithm is optimal (up to a logarithmic factor) for the decision classes derived from matroids (including TopK and MB), while for general decision classes, our upper and lower bounds are within a relatively benign factor.

2.8 Analysis of CLUCB (Theorem 2.1)

In this section, we analyze the sample complexity of CLUCB and prove Theorem 2.1.

Notations. Fix some decision class $\mathcal{M} \subseteq 2^{[n]}$ and fix some expected reward vector $\boldsymbol{w} \in \mathbb{R}^n$. Recall that $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ is the optimal set. Since we assume that M_* is unique, one can verify that, for every $e \in [n]$, the gap defined in Eq. (2.1) is positive, i.e., $\Delta_e > 0$.

We will also need some additional notations for our analysis. For any set $a \subseteq [n]$, let $\boldsymbol{\chi}_a \in \{0,1\}^n$ denote the incidence vector of set $a \subseteq [n]$, i.e., $\chi_a(e) = 1$ if and only if $e \in a$. For an exchange set $b = (b_+, b_-)$, we define $\boldsymbol{\chi}_b \triangleq \boldsymbol{\chi}_{b_+} - \boldsymbol{\chi}_{b_-}$ as the incidence vector of b. We notice that $\boldsymbol{\chi}_b \in \{-1, 0, 1\}^n$.

For each round t, we define vector $\mathbf{rad}_t = \big(\operatorname{rad}_t(1), \ldots, \operatorname{rad}_t(n)\big)^T$ and recall that $\bar{\boldsymbol{w}}_t \in \mathbb{R}^n$ is the empirical mean rewards of arms up to round t.

Let $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^n$ be two vectors. Let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ denote the inner product of $\boldsymbol{u}$ and $\boldsymbol{v}$. We define $\boldsymbol{u} \circ \boldsymbol{v} \triangleq \big(u(1) \cdot v(1), \ldots, u(n) \cdot v(n)\big)^T$ as the element-wise product of $\boldsymbol{u}$ and $\boldsymbol{v}$. For any $s \in \mathbb{R}$, we also define $\boldsymbol{u}^s \triangleq \big(u(1)^s, \ldots, u(n)^s\big)^T$ as the element-wise exponentiation of $\boldsymbol{u}$. We let $|\boldsymbol{u}| = \big(|u(1)|, \ldots, |u(n)|\big)^T$ denote the element-wise absolute value of $\boldsymbol{u}$.

Finally, let us recall that for any exchange class $b = (b_+, b_-)$ and any set $M \subseteq [n]$, we have defined $M \oplus b = M \backslash b_- \cup b_+$ and $M \ominus b = M \backslash b_+ \cup b_-$.

2.8.1 Preparatory Lemmas

Let us begin with a simple lemma that characterizes the incidence vectors of exchange sets.

Lemma 2.1. *Let $M_1 \subseteq [n]$ be a set. Let $b = (b_+, b_-)$ be an exchange set such that $b_- \subseteq M_1$ and $b_+ \cap M_1 = \emptyset$. Define $M_2 = M_1 \oplus b$. Then, we have*

$$\boldsymbol{\chi}_{M_1} + \boldsymbol{\chi}_b = \boldsymbol{\chi}_{M_2}.$$

In addition, we have $M_1 = M_2 \ominus b$.

Proof. Recall that $M_2 = M_1 \backslash b_- \cup b_+$ and $b_+ \cap b_- = \emptyset$. Therefore we see that $M_2 \backslash M_1 = b_+$ and $M_1 \backslash M_2 = b_-$. We can decompose $\boldsymbol{\chi}_{M_1}$ as $\boldsymbol{\chi}_{M_1} = \boldsymbol{\chi}_{M_1 \backslash M_2} + \boldsymbol{\chi}_{M_1 \cap M_2}$. Hence, we have

$$\boldsymbol{\chi}_{M_1} + \boldsymbol{\chi}_b = \boldsymbol{\chi}_{M_1 \backslash M_2} + \boldsymbol{\chi}_{M_1 \cap M_2} + \boldsymbol{\chi}_{b_+} - \boldsymbol{\chi}_{b_-}$$

$$
\begin{aligned}
&= \boldsymbol{\chi}_{M_1 \cap M_2} + \boldsymbol{\chi}_{M_2 \backslash M_1} \\
&= \boldsymbol{\chi}_{M_2}.
\end{aligned}
$$

Using the definition of operator $\ominus$, one can verify that $M_1 = M_2 \ominus b$. □

The next lemma serves as a basic tool derived from exchange classes, which allows us to interpolate between different members of a decision class $\mathcal{M}$. Moreover, it characterizes the relationship between gaps and exchange sets. In Figure 2.1, we illustrate the intuitions of the interpolations characterized in Lemma 2.2.

Lemma 2.2 (Interpolation Lemma)**.** *Let $\mathcal{M} \subseteq 2^{[n]}$ and let $\mathcal{B}$ be an exchange class for $\mathcal{M}$. Then, for any two different members M, M' of $\mathcal{M}$ and any $e \in (M \backslash M') \cup (M' \backslash M)$, there exists an exchange set $b = (b_+, b_-) \in \mathcal{B}$ which satisfies five constraints:* ***(a)*** *$e \in (b_+ \cup b_-)$,* ***(b)*** *$b_- \subseteq (M \backslash M')$,* ***(c)*** *$b_+ \subseteq (M' \backslash M)$,* ***(d)*** *$(M \oplus b) \in \mathcal{M}$ and* ***(e)*** *$(M' \ominus b) \in \mathcal{M}$. Moreover, if $M' = M_*$, then we have $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e$, where Δ_e is the gap defined in Eq.* (2.1)*.*

Proof. We decompose our proof into two cases.

Case (1): $e \in M \backslash M'$.

By the definition of exchange class, we know that there exists $b = (b_+, b_-) \in \mathcal{B}$ which satisfies that $e \in b_-$, $b_- \subseteq (M \backslash M')$, $b_+ \subseteq (M' \backslash M)$, $(M \oplus b) \in \mathcal{M}$ and $(M' \ominus b) \in \mathcal{M}$. Therefore the five constraints are satisfied.

Next, if $M' = M_*$, we see that $e \notin M_*$. Let us consider the set $M_1 = \arg\max_{S \in \mathcal{M} \colon e \in S} w(S)$. Note that, by the definition of gaps (Eq. (2.1)), one has $w(M_*) - w(M_1) = \Delta_e$. Now we define $M_0 = M_* \ominus b$. Note that we already have $M_* \ominus b \in \mathcal{M}$. By combining this with the fact that $e \in M_0$, we see that $w(M_0) \leq \max_{S \in \mathcal{M} \colon e \in S} w(S) = w(M_1)$. Therefore, we obtain that $w(M_*) - w(M_0) \geq w(M_*) - w(M_1) = \Delta_e$. Notice that the left-hand side of the former inequality can be rewritten using Lemma 2.1 as follows

$$\begin{aligned} w(M_*) - w(M_0) &= \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_*} \right\rangle - \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_0} \right\rangle \\ &= \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_*} - \boldsymbol{\chi}_{M_0} \right\rangle \\ &= \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{b} \right\rangle . \end{aligned}$$

Therefore, we obtain $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e$.

Case (2): $e \in M' \backslash M$**.**

Using the definition of exchange class, we see that there exists $c = (c_+, c_-) \in \mathcal{B}$ such that $e \in c_-$, $c_- \subseteq (M' \backslash M)$, $c_+ \subseteq (M \backslash M')$, $(M' \oplus c) \in \mathcal{M}$ and $(M \ominus c) \in \mathcal{M}$.

We construct $b = (b_+, b_-)$ by setting $b_+ = c_-$ and $b_- = c_+$. Notice that, by the construction of b, we have $M \oplus b = M \ominus c$ and $M' \ominus b = M' \oplus c$. Therefore, it is clear that b satisfies the five constraints of the lemma.

Now, suppose that $M' = M_*$. In this case, we have $e \in M_*$. Consider the set $M_3 = \arg\max_{S \in \mathcal{M} \colon e \notin S} w(S)$. By definition of Δ_e, we see that $w(M_*) - w(M_3) = \Delta_e$. Now we define $M_2 = M_* \ominus b$ and notice that $M_2 \in \mathcal{M}$. By combining with the fact that $e \notin M_2$, we obtain that $w(M_2) \leq \max_{S \in \mathcal{M} \colon e \notin S} = w(M_3)$. Hence, we have $w(M_*) - w(M_2) \geq w(M_*) - w(M_3) = \Delta_e$. Similar to Case (1), applying Lemma 2.1 again, we have

$$\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle = w(M_*) - w(M_2) \geq \Delta_e.$$

□

Next we state two basic lemmas that help us to convert set-theoretical arguments to linear algebraic arguments.

Lemma 2.3. *Let $M, M' \subseteq [n]$ be two sets. Let $\mathbf{rad}_t$ be an n-dimensional vector with nonnegative entries. Then, we have*

$$\max_{e \in (M \backslash M') \cup (M' \backslash M)} \mathrm{rad}_t(e) = \left\| \mathbf{rad}_t \circ \left| \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_M \right| \right\|_\infty .$$

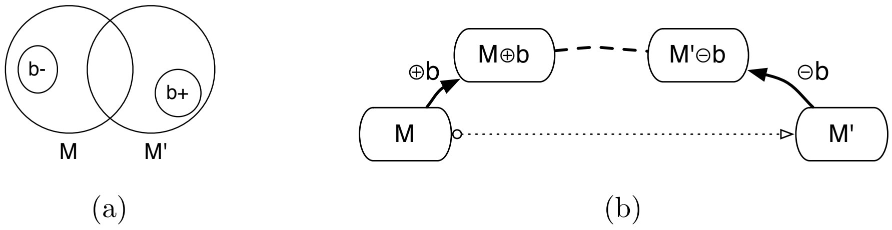


Figure 2.1: Illustration of Lemma 2.2: (a) A Venn diagram for the relationships among M, M', b_- and b_+. Note that $e \in b_- \cup b_+$. (b) An illustration for the relationships among M, M', $M \oplus b$ and $M' \ominus b$. We recall that $M \oplus b = M \backslash b_- \cup b_+$ and $M' \ominus b = M' \backslash b_+ \cup b_-$. We use dotted line to represent an application of Lemma 2.2 between two sets.

Proof. Notice that $\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_M = \boldsymbol{\chi}_{M'\backslash M} - \boldsymbol{\chi}_{M\backslash M'}$. In addition, since $(M'\backslash M) \cap (M\backslash M') = \emptyset$, we have $\boldsymbol{\chi}_{M'\backslash M} \circ \boldsymbol{\chi}_{M\backslash M'} = \mathbf{0}_n$. Also notice that $\boldsymbol{\chi}_{M'\backslash M} - \boldsymbol{\chi}_{M\backslash M'} \in \{-1, 0, 1\}^n$. Therefore, we have

$$\begin{aligned}|\boldsymbol{\chi}_{M'\backslash M} - \boldsymbol{\chi}_{M\backslash M'}| &= (\boldsymbol{\chi}_{M'\backslash M} - \boldsymbol{\chi}_{M\backslash M'})^2 \\ &= \boldsymbol{\chi}^2_{M'\backslash M} + \boldsymbol{\chi}^2_{M\backslash M'} - 2\boldsymbol{\chi}_{M'\backslash M} \circ \boldsymbol{\chi}_{M\backslash M'} \\ &= \boldsymbol{\chi}_{M'\backslash M} + \boldsymbol{\chi}_{M\backslash M'} \\ &= \boldsymbol{\chi}_{(M'\backslash M)\cup(M\backslash M')},\end{aligned}$$

where the third equation follows from the fact that $\boldsymbol{\chi}_{M\backslash M'} \in \{0,1\}^n$ and $\boldsymbol{\chi}_{M'\backslash M} \in \{0,1\}^n$. The lemma follows immediately from the fact that $\mathrm{rad}_t(e) \geq 0$ and $\boldsymbol{\chi}_{(M\backslash M')\cup(M'\backslash M)} \in \{0,1\}^n$. □

Lemma 2.4. *Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^n$ be three vectors. Then, we have $\langle \boldsymbol{a}, \boldsymbol{b} \circ \boldsymbol{c} \rangle = \langle \boldsymbol{a} \circ \boldsymbol{b}, \boldsymbol{c} \rangle$.*

Proof. We have

$$\langle \boldsymbol{a}, \boldsymbol{b} \circ \boldsymbol{c} \rangle = \sum_{i=1}^{n} a(i)\big(b(i)c(i)\big) = \sum_{i=1}^{n} \big(a(i)b(i)\big)c(i) = \langle \boldsymbol{a} \circ \boldsymbol{b}, \boldsymbol{c} \rangle .$$

□

The next lemma characterizes the property of $\tilde{\boldsymbol{w}}_t$ which is defined in the CLUCB algorithm.

Lemma 2.5. *Let M_t, $\tilde{\boldsymbol{w}}_{\boldsymbol{t}}$ and $\mathbf{rad}_t$ be defined in Algorithm 1 and Theorem 2.1. Let $M' \in \mathcal{M}$ be an arbitrary member of decision class. We have*

$$\begin{aligned}\tilde{w}_t(M') - \tilde{w}_t(M_t) &= \left\langle \tilde{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle \\ &= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t}| \right\rangle.\end{aligned}$$

Proof. We begin with proving the first part. It is easy to verify that $\tilde{\boldsymbol{w}}_{\boldsymbol{t}} = \bar{\boldsymbol{w}}_t + \mathbf{rad}_t \circ (\mathbf{1}_n - 2\boldsymbol{\chi}_{M_t})$. Then, we have

$$\begin{aligned}&\left\langle \tilde{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle \\ =&\left\langle \bar{\boldsymbol{w}}_t + \mathbf{rad}_t \circ (\mathbf{1}_n - 2\boldsymbol{\chi}_{M_t}),\ \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle \\ =&\left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, (\mathbf{1}_n - 2\boldsymbol{\chi}_{M_t}) \circ (\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t}) \right\rangle \\ &\hfill (2.9) \\ =&\left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} - 2\boldsymbol{\chi}_{M_t} \circ \boldsymbol{\chi}_{M'} + 2\boldsymbol{\chi}_{M_t}^2 \right\rangle \\ =&\left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, \boldsymbol{\chi}_{M'}^2 - \boldsymbol{\chi}_{M_t}^2 - 2\boldsymbol{\chi}_{M_t} \circ \boldsymbol{\chi}_{M'} + 2\boldsymbol{\chi}_{M_t}^2 \right\rangle \\ &\hfill (2.10) \\ =&\left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, (\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t})^2 \right\rangle \\ =&\left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t}| \right\rangle, \qquad (2.11)\end{aligned}$$

where Eq. (2.9) follows from Lemma 2.4; Eq. (2.10) holds since $\boldsymbol{\chi}_{M'} \in \{0,1\}^n$ and $\boldsymbol{\chi}_{M_t} \in \{0,1\}^n$ and therefore $\boldsymbol{\chi}_{M'} = \boldsymbol{\chi}_{M'}^2$ and $\boldsymbol{\chi}_{M_t} = \boldsymbol{\chi}_{M_t}^2$; and Eq. (2.11) follows since $\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t} \in \{-1,0,1\}^n$. □

2.8.2 Confidence Intervals

First, we recall a standard concentration inequality of sub-Gaussian random variables.

Lemma 2.6 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be n independent random variables such that, for each $i \in [n]$, random variable $X_i - \mathbb{E}[X_i]$ is R-sub-Gaussian distributed, i.e.,*

$\forall t \in \mathbb{R}$, $\mathbb{E}[\exp(tX_i - t\mathbb{E}[X_i])] \le \exp(R^2t^2/2)$. Let $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ denote the average of these random variables. Then, for any $\lambda > 0$, we have

$$\Pr\left[\left|\bar{X} - \mathbb{E}[\bar{X}]\right| \ge \lambda\right] \le 2\exp\left(-\frac{n\lambda^2}{2R^2}\right).$$

Proof. For all $i \in [n]$, we define $v_i = X_i - \mathbb{E}[X_i]$. We also define $S = \sum_{i=1}^n v_i$ and $\epsilon = n\lambda$. Therefore, for any $t > 0$, we have

$$\begin{aligned}\Pr[S \ge \epsilon] = \Pr[tS \ge t\epsilon] &\overset{\text{(a)}}{\le} \frac{\mathbb{E}[\exp(tS)]}{\exp(t\epsilon)} = \frac{\mathbb{E}[\exp(\sum_{i=1}^n tv_i)]}{\exp(t\epsilon)} \\ &\overset{\text{(b)}}{=} \frac{\prod_{i=1}^n \mathbb{E}[\exp(tv_i)]}{\exp(t\epsilon)} \le \frac{\prod_{i=1}^n \exp\left(R^2t^2/2\right)}{\exp(t\epsilon)} \\ &= \exp(nR^2t^2/2 - t\epsilon),\end{aligned}$$

where (a) follows from Markov's inequality and (b) holds since $v_1, \ldots, v_n$ are independent. Now minimizing over $t > 0$, we get

$$\begin{aligned}\Pr[S \ge \epsilon] &\le \inf_{t>0} \exp(nR^2t^2/2 - t\epsilon) \\ &= \exp(-\epsilon^2/2nR^2) \\ &= \exp(-n\lambda^2/2R^2).\end{aligned}$$

Similarly, one can show that $\Pr[S \le -\epsilon] \le \exp(-n\lambda^2/2R^2)$. Hence, the lemma follows from a union bound. □

Next, for all $t > 0$, we define random event ξ_t as follows

$$\xi_t = \left\{\forall i \in [n], \quad |w(i) - \bar{w}_t(i)| < \mathrm{rad}_t(i)\right\}. \tag{2.12}$$

We notice that random event ξ_t characterizes the event that the confidence bounds of all arms are valid at round t.

If the confidence bounds are valid, we can generalize Eq. (2.12) to inner products as follows.

Lemma 2.7. *Given any $t > 0$, assume that event ξ_t as defined in Eq. (2.12) occurs. Then, for any vector $\boldsymbol{a} \in \mathbb{R}^n$, we have*

$$\left|\,\langle \boldsymbol{w}, \boldsymbol{a}\rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{a}\rangle\,\right| < \langle \mathbf{rad}_t, |\boldsymbol{a}|\rangle.$$

Proof. Suppose that ξ_t occurs. Then, we have

$$\begin{aligned}
\big|\, \langle \boldsymbol{w}, \boldsymbol{a} \rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{a} \rangle \,\big| &= \big|\, \langle \boldsymbol{w} - \bar{\boldsymbol{w}}_t, \boldsymbol{a} \rangle \,\big| \\
&= \left| \sum_{i=1}^{n} \big(w(i) - \bar{w}_t(i) \big) a(i) \right| \\
&\le \sum_{i=1}^{n} \big| w(i) - \bar{w}_t(i) \big| |a(i)| \\
&< \sum_{i=1}^{n} \mathrm{rad}_t(i) \cdot |a(i)| \qquad (2.13) \\
&= \langle \mathbf{rad}_t, |\boldsymbol{a}| \rangle ,
\end{aligned}$$

where Eq. (2.13) follows the definition of event ξ_t in Eq. (2.12) and the assumption that it occurs. □

Next, we construct the high probability confidence intervals for the fixed confidence setting.

Lemma 2.8. *Suppose that the reward distribution φ_e is a R-sub-Gaussian distribution for all $e \in [n]$. And if, for all $t > 0$ and all $e \in [n]$, the confidence radius* $\mathrm{rad}_t(e)$ *is given by*

$$\mathrm{rad}_t(e) = R \sqrt{\frac{2 \log \left(\frac{4nt^3}{\delta} \right)}{T_t(e)}},$$

where $T_t(e)$ is the number of samples of arm e up to round t. Then, we have

$$\Pr \left[\bigcap_{t=1}^{\infty} \xi_t \right] \ge 1 - \delta.$$

Proof. Fix any $t > 0$ and $e \in [n]$. Note that φ_e is a R-sub-Gaussian tail distribution with mean $w(e)$ and $\bar{w}_t(e)$ is the empirical mean of φ_e from $T_t(e)$ samples. Then, we have

$$\Pr \left[\big| \bar{w}_t(e) - w(e) \big| \ge R \sqrt{\frac{2 \log \left(\frac{4nt^3}{\delta} \right)}{T_t(e)}} \right]$$

$$
\begin{aligned}
&= \sum_{s=1}^{t-1} \Pr\left[\left| \bar{w}_t(e) - w(e) \right| \geq R\sqrt{\frac{2\log\left(\frac{4nt^3}{\delta}\right)}{s}}, T_t(e) = s \right] && (2.14)\\
&\leq \sum_{s=1}^{t-1} \frac{\delta}{2nt^3} && (2.15)\\
&\leq \frac{\delta}{2nt^2},
\end{aligned}
$$

where Eq. (2.14) follows from the fact that $1 \leq T_t(e) \leq t-1$ and Eq. (2.15) follows from Hoeffding's inequality (Lemma 2.6). By a union bound over all $e \in [n]$, we see that $\Pr[\xi_t] \geq 1 - \frac{\delta}{2t^2}$. Using a union bound again over all $t > 0$, we have

$$
\begin{aligned}
\Pr\left[\bigcap_{t=1}^{\infty} \xi_t \right] &\geq 1 - \sum_{t=1}^{\infty} \Pr[\neg \xi_t]\\
&\geq 1 - \sum_{t=1}^{\infty} \frac{\delta}{2t^2}\\
&= 1 - \frac{\pi^2}{12}\delta \geq 1 - \delta.
\end{aligned}
$$

□

2.8.3 Main Lemmas

Now we state our key technical lemmas. In these lemmas, we shall use Lemma 2.2 to construct gadgets that interpolate between different members of a decision class. The first lemma shows that, if the confidence intervals are valid, then CLUCB always returns the correct answer when it stops.

Lemma 2.9. *Given any $t > n$, assume that event ξ_t (defined in Eq. (2.12)) occurs. Then, if Algorithm 1 terminates at round t, we have $M_t = M_*$.*

Proof. Suppose that $M_t \neq M_*$. By the assumption that M_* is the unique optimal set, we have $w(M_*) > w(M_t)$. Rewriting this inequality, we obtain that $\left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_*} \right\rangle > \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_t} \right\rangle$.

Let $\mathcal{B}$ be an exchange class for $\mathcal{M}$. Applying Lemma 2.2 by setting $M = M_t$ and $M' = M_*$, we see that there exists $b = (b_+, b_-) \in \mathcal{B}$ such that $(M_t \oplus b) \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle > 0$.

Now define $M_t' = M_t \oplus b$. Recall that $\tilde{M}_t = \arg\max_{M \in \mathcal{M}} \tilde{w}_t(M)$ and therefore $\tilde{w}_t(\tilde{M}_t) \geq \tilde{w}_t(M_t')$. Hence, we have

$$\begin{aligned}
\tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) &\geq \tilde{w}_t(M_t') - \tilde{w}_t(M_t) \\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t'} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{M'} - \boldsymbol{\chi}_{M_t}| \right\rangle \qquad (2.16) \\
&\geq \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_t'} - \boldsymbol{\chi}_{M_t} \right\rangle \qquad (2.17) \\
&= \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle > 0, \qquad (2.18)
\end{aligned}$$

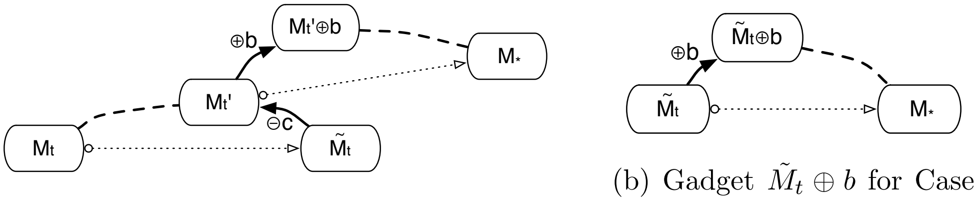
where Eq. (2.16) follows from Lemma 2.5; and Eq. (2.17) follows the assumption that event ξ_t occurs and Lemma 2.7.

Therefore Eq. (2.18) shows that $\tilde{w}_t(\tilde{M}_t) > \tilde{w}_t(M_t)$. However, this contradicts to the stopping condition of CLUCB: $\tilde{w}_t(\tilde{M}_t) = \tilde{w}_t(M_t)$ and the assumption that the algorithm terminates on round t. □

The next lemma shows that if the confidence interval of an arm is sufficiently small, then this arm will not be played by the algorithm. In the proof, we construct a number of gadgets using Lemma 2.2. We illustrate the relationships among the gadgets in Figure 2.2.

Lemma 2.10. *Given any $t > 0$ and suppose that event ξ_t (defined in Eq. (2.12)) occurs. For any $e \in [n]$, if $\mathrm{rad}_t(e) < \frac{\Delta_e}{3 width(\mathcal{M})}$, then, arm e will not be pulled on round t, i.e., $p_t \neq e$.*

Proof. We prove by contradiction. Therefore we shall assume the opposite that $p_t = e$ in the rest of the proof.



(a) Gadgets M_t' and $M_t' \oplus b$ for Case (1)

(b) Gadget $\tilde{M}_t \oplus b$ for Case (2)

Figure 2.2: An illustration of the relationship among the gadgets used the proof of Lemma 2.10; We use dotted line to represent an application of Lemma 2.2 between two sets.

First let us fix an exchange class $\mathcal{B} \in \arg\min_{\mathcal{B}' \in \mathrm{Exchange}(\mathcal{M})} \mathrm{width}(\mathcal{B}')$. Note that $\mathrm{width}(\mathcal{B}) = \mathrm{width}(\mathcal{M})$. By Lemma 2.2, there exists an exchange set $c = (c_+, c_-) \in \mathcal{B}$ such that $e \in (c_+ \cup c_-)$, $c_- \subseteq (M_t \backslash \tilde{M}_t)$, $c_+ \subseteq (\tilde{M}_t \backslash M_t)$, $(M_t \oplus c) \in \mathcal{M}$ and $(\tilde{M}_t \ominus c) \in \mathcal{M}$.

Now, we decompose our proof into two cases.

Case (1): $(e \in M_* \wedge e \in c_+) \vee (e \notin M_* \wedge e \in c_-)$**.**

First we construct a gadget $M_t' = \tilde{M}_t \ominus c$ and recall that $M_t' \in \mathcal{M}$. By the definitions of $\ominus$ and $\oplus$, we see that $\tilde{M}_t = M_t' \oplus c$.

We claim that $M_t' \neq M_*$. The assumption of Case (1) means that either (a) $e \in M_*$ and $e \in c_+$; or (b) $e \notin M_*$ and $e \in c_-$ holds. Suppose that $e \in M_*$ and $e \in c_+$. Then, we see that $e \notin M_t'$ and hence $M_t' \neq M_*$. On the other hand, if $e \notin M_*$ and $e \in c_-$, then $e \in M_t'$ which also means that $M_t' \neq M_*$. Therefore we have $M_t' \neq M_*$ in either cases.

Next, we apply Lemma 2.2 by setting $M = M_t'$ and $M' = M_*$. We see that there exists an exchange set $b \in \mathcal{B}$ such that, $e \in (b_+ \cup b_-)$, $(M_t' \oplus b) \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e > 0$. We will also use $M_t' \oplus b$ as a gadget.

Now, we define vectors $\boldsymbol{d} = \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t}$, $\boldsymbol{d}_1 = \boldsymbol{\chi}_{M_t'} - \boldsymbol{\chi}_{M_t}$ and $\boldsymbol{d}_2 = \boldsymbol{\chi}_{M_t' \oplus b} - \boldsymbol{\chi}_{M_t}$. By the definition of M_t' and Lemma 2.1, we see that $\boldsymbol{d}_1 = \boldsymbol{d} - \boldsymbol{\chi}_c$ and $\boldsymbol{d}_2 = \boldsymbol{d}_1 + \boldsymbol{\chi}_b = \boldsymbol{d} - \boldsymbol{\chi}_c + \boldsymbol{\chi}_b$.

Then, we claim that $\|\mathbf{rad}_t \circ (\boldsymbol{d} - \boldsymbol{\chi}_c)\|_\infty < \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}$. To prove this claim, we first appeal to standard set-algebraic manipulations. We obtain

$$\begin{aligned}
M_t \backslash M'_t &= M_t \backslash (\tilde{M}_t \ominus c) \\
&= M_t \backslash (\tilde{M}_t \backslash c_+ \cup c_-) \\
&= M_t \backslash (\tilde{M}_t \backslash c_+) \cap (M_t \backslash c_-) \\
&= (M_t \cap c_+) \cup (M_t \backslash \tilde{M}_t) \cap (M_t \backslash c_-) \\
&= (M_t \backslash \tilde{M}_t) \cap (M_t \backslash c_-) && (2.19) \\
&\subseteq M_t \backslash \tilde{M}_t, && (2.20)
\end{aligned}$$

where Eq. (2.19) follows from $c_+ \subseteq \tilde{M}_t \backslash M_t$ and therefore $c_+ \cap M_t = \emptyset$. Similarly, we can derive $M'_t \backslash M_t$ as follows

$$\begin{aligned}
M'_t \backslash M_t &= (\tilde{M}_t \ominus c) \backslash M_t = (\tilde{M}_t \backslash c_+ \cup c_-) \backslash M_t \\
&= \big((\tilde{M}_t \backslash c_+) \backslash M_t\big) \cup (c_- \backslash M_t) \\
&= \tilde{M}_t \backslash c_+ \backslash M_t && (2.21) \\
&\subseteq \tilde{M}_t \backslash M_t, && (2.22)
\end{aligned}$$

where Eq. (2.21) follows from $c_- \subseteq M_t \backslash \tilde{M}_t$ and hence $c_- \backslash M_t = \emptyset$. By combining Eq. (2.20) and Eq. (2.22), we see that $\big((M_t \backslash M'_t) \cup (M'_t \backslash M_t)\big) \subseteq \big((M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)\big)$. Then, applying Lemma 2.3, we obtain

$$\begin{aligned}
\|\mathbf{rad}_t \circ (\boldsymbol{d} - \boldsymbol{\chi}_c)\|_\infty &= \big\|\mathbf{rad}_t \circ (\boldsymbol{\chi}_{M'_t} - \boldsymbol{\chi}_{M_t})\big\|_\infty \\
&= \max_{i \in (M_t \backslash M'_t) \cup (M'_t \backslash M_t)} \mathrm{rad}_t(i) \\
&\le \max_{i \in (M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)} \mathrm{rad}_t(i) \\
&= \mathrm{rad}_t(e) && (2.23) \\
&< \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}, && (2.24)
\end{aligned}$$

where Eq. (2.23) follows from the assumption that $p_t = e$.

Next we claim that $\|\mathbf{rad}_t \circ \boldsymbol{\chi}_c\|_\infty < \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}$. Recall that, by the definition of c, we have $c_+ \subseteq (\tilde{M}_t \backslash M_t)$ and $c_- \subseteq (M_t \backslash \tilde{M}_t)$.

Hence $c_+ \cup c_- \subseteq (\tilde{M}_t \backslash M_t) \cup (M_t \backslash \tilde{M}_t)$. Since $\boldsymbol{\chi}_c \in \{-1, 0, 1\}^n$, we see that

$$\begin{aligned} \|\mathbf{rad}_t \circ \boldsymbol{\chi}_c\|_\infty &= \max_{i \in c_+ \cup c_-} \mathrm{rad}_t(i) \\ &\leq \max_{i \in (\tilde{M}_t \backslash M_t) \cup (M_t \backslash \tilde{M}_t)} \mathrm{rad}_t(i) \\ &= \mathrm{rad}_t(e) < \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}. \end{aligned} \tag{2.25}$$

From Eq. (2.25), we derive

$$\begin{aligned} \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle &= \left\langle \mathbf{rad}_t, \boldsymbol{\chi}_c^2 \right\rangle && (2.26) \\ &= \langle \mathbf{rad}_t \circ \boldsymbol{\chi}_c, \boldsymbol{\chi}_c \rangle && (2.27) \\ &\leq \|\mathbf{rad}_t \circ \boldsymbol{\chi}_c\|_\infty \|\boldsymbol{\chi}_c\|_1 && (2.28) \\ &< \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})} \|\boldsymbol{\chi}_c\|_1 && (2.29) \\ &\leq \frac{\Delta_e}{3}, && (2.30) \end{aligned}$$

where Eq. (2.26) hold since $\boldsymbol{\chi}_c \in \{-1, 0, 1\}^n$; Eq. (2.27) follows form Lemma 2.4; Eq. (2.28) follows from Hölder's inequality; Eq. (2.29) follows from Eq. (2.25); and Eq. (2.30) holds since $\|\boldsymbol{\chi}_c\|_1 = |c_+| + |c_-| \leq \mathrm{width}(\mathcal{B})$ where the inequality is due to $c \in \mathcal{B}$.

Next, we claim that $\boldsymbol{d} \circ \boldsymbol{\chi}_c = |\boldsymbol{\chi}_c|$. Recall that $\boldsymbol{\chi}_c = \boldsymbol{\chi}_{c_+} - \boldsymbol{\chi}_{c_-}$ and $\boldsymbol{d} = \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t} = \boldsymbol{\chi}_{\tilde{M}_t \backslash M_t} - \boldsymbol{\chi}_{M_t \backslash \tilde{M}_t}$. We also notice that $c_+ \subseteq (\tilde{M}_t \backslash M_t)$ and $c_- \subseteq (M_t \backslash \tilde{M}_t)$. This implies that $c_+ \cap (M_t \backslash \tilde{M}_t) = \emptyset$ and $c_- \cap (\tilde{M}_t \backslash M_t) = \emptyset$. Therefore, we have

$$\begin{aligned} \boldsymbol{d} \circ \boldsymbol{\chi}_c &= (\boldsymbol{\chi}_{\tilde{M}_t \backslash M_t} - \boldsymbol{\chi}_{M_t \backslash \tilde{M}_t}) \circ (\boldsymbol{\chi}_{c_+} - \boldsymbol{\chi}_{c_-}) \\ &= \boldsymbol{\chi}_{\tilde{M}_t \backslash M_t} \circ \boldsymbol{\chi}_{c_+} + \boldsymbol{\chi}_{M_t \backslash \tilde{M}_t} \circ \boldsymbol{\chi}_{c_-} - \boldsymbol{\chi}_{\tilde{M}_t \backslash M_t} \circ \boldsymbol{\chi}_{c_-} - \boldsymbol{\chi}_{M_t \backslash \tilde{M}_t} \circ \boldsymbol{\chi}_{c_+} \\ &= \boldsymbol{\chi}_{\tilde{M}_t \backslash M_t} \circ \boldsymbol{\chi}_{c_+} + \boldsymbol{\chi}_{M_t \backslash \tilde{M}_t} \circ \boldsymbol{\chi}_{c_-} \\ &= \boldsymbol{\chi}_{c_+} + \boldsymbol{\chi}_{c_-} = |\boldsymbol{\chi}_c|. \end{aligned}$$

where the second equality holds since $c_+ \cap (M_t \backslash \tilde{M}_t) = \emptyset$ and $c_- \cap (\tilde{M}_t \backslash M_t) = \emptyset$; and the last equality holds since $c_+ \cap c_- = \emptyset$.

Now, we bound quantity $\langle \mathbf{rad}_t, |\boldsymbol{d}_2| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle$ as follows

$$\begin{aligned}
&\langle \mathbf{rad}_t, |\boldsymbol{d}_2| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle \\
=&\langle \mathbf{rad}_t, |\boldsymbol{d}_2| - |\boldsymbol{d}| \rangle = \left\langle \mathbf{rad}_t, \boldsymbol{d}_2^2 - \boldsymbol{d}^2 \right\rangle && (2.31)\\
=&\left\langle \mathbf{rad}_t, (\boldsymbol{d} - \boldsymbol{\chi}_c + \boldsymbol{\chi}_b)^2 - \boldsymbol{d}^2 \right\rangle \\
=&\left\langle \mathbf{rad}_t, \boldsymbol{\chi}_b^2 + \boldsymbol{\chi}_c^2 - 2\boldsymbol{\chi}_b \circ \boldsymbol{\chi}_c - 2\boldsymbol{d} \circ \boldsymbol{\chi}_c + 2\boldsymbol{d} \circ \boldsymbol{\chi}_b \right\rangle \\
=&\left\langle \mathbf{rad}_t, \boldsymbol{\chi}_b^2 - \boldsymbol{\chi}_c^2 + 2\boldsymbol{\chi}_b \circ (\boldsymbol{d} - \boldsymbol{\chi}_c) \right\rangle && (2.32)\\
=&\langle \mathbf{rad}_t, |\boldsymbol{\chi}_b| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle - 2 \langle \mathbf{rad}_t, \boldsymbol{\chi}_b \circ (\boldsymbol{d} - \boldsymbol{\chi}_c) \rangle \\
=&\langle \mathbf{rad}_t, |\boldsymbol{\chi}_b| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle - 2 \langle \mathbf{rad}_t \circ (\boldsymbol{d} - \boldsymbol{\chi}_c), \boldsymbol{\chi}_b \rangle && (2.33)\\
\geq&\langle \mathbf{rad}_t, |\boldsymbol{\chi}_b| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle - 2 \left\| \mathbf{rad}_t \circ (\boldsymbol{d} - \boldsymbol{\chi}_c) \right\|_\infty \left\| \boldsymbol{\chi}_b \right\|_1 \\
& && (2.34)\\
>&\langle \mathbf{rad}_t, |\boldsymbol{\chi}_b| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle - \frac{2\Delta_e}{3\mathrm{width}(\mathcal{B})} \| \boldsymbol{\chi}_b \|_1 && (2.35)\\
\geq&\langle \mathbf{rad}_t, |\boldsymbol{\chi}_b| \rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c| \rangle - \frac{2\Delta_e}{3}, && (2.36)
\end{aligned}$$

where Eq. (2.31) holds since $\boldsymbol{d} \in \{-1, 0, 1\}^n$ and $\boldsymbol{d}_2 \in \{-1, 0, 1\}^n$; Eq. (2.32) follows from the claim that $\boldsymbol{d} \circ \boldsymbol{\chi}_c = |\boldsymbol{\chi}_c| = \boldsymbol{\chi}_c^2$; Eq. (2.33) and Eq. (2.34) follow from Lemma 2.4 and Hölder's inequality; Eq. (2.35) follows from Eq. (2.24); and Eq. (2.36) holds since $b \in \mathcal{B}$ and $\| \boldsymbol{\chi}_b \|_1 = |b_+| + |b_-| \leq \mathrm{width}(\mathcal{B})$.

Applying Lemma 2.5 by setting $M' = \tilde{M}_t$, we have

$$\begin{aligned}
\langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle &= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t}| \right\rangle \\
&= \tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \\
&\geq \tilde{w}_t(M'_t \oplus b) - \tilde{w}_t(M_t) && (2.37)\\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'_t \oplus b} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{M'_t \oplus b} - \boldsymbol{\chi}_{M_t}| \right\rangle \\
&= \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d}_2 \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_2| \rangle \\
&= \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_c \rangle + \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_2| \rangle, \\
& && (2.38)
\end{aligned}$$

where Eq. (2.37) follows from the fact that $\tilde{w}_t(\tilde{M}_t) = \max_{M\in\mathcal{M}} \tilde{w}_t(M)$; and Eq. (2.38) follows from the fact that $\boldsymbol{d}_2 = \boldsymbol{d} - \boldsymbol{\chi}_c + \boldsymbol{\chi}_b$. Rearranging the above inequality, we obtain

$$\begin{aligned}
\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_c\rangle &\geq \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b\rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_2|\rangle - \langle \mathbf{rad}_t, |\boldsymbol{d}|\rangle \\
&\geq \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b\rangle + \langle \mathbf{rad}_t, |\boldsymbol{\chi}_b|\rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c|\rangle - \frac{2\Delta_e}{3} && (2.39)\\
&> \langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle - \langle \mathbf{rad}_t, |\boldsymbol{\chi}_c|\rangle - \frac{2\Delta_e}{3} && (2.40)\\
&> \langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle - \frac{\Delta_e}{3} - \frac{2\Delta_e}{3} && (2.41)\\
&= \langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle - \Delta_e \geq 0, && (2.42)
\end{aligned}$$

where Eq. (2.39) uses Eq. (2.36); Eq. (2.40) follows from the assumption that event ξ_t occurs and Lemma 2.7; and Eq. (2.41) holds due to Eq. (2.30).

We have shown that $\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_c\rangle > 0$. Now we can bound $\bar{w}_t(M'_t)$ as follows

$$\begin{aligned}
\bar{w}_t(M'_t) &= \big\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M'_t}\big\rangle \\
&= \big\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} + \boldsymbol{\chi}_c\big\rangle \\
&= \big\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t}\big\rangle + \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_c\rangle \\
&> \big\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t}\big\rangle \\
&= \bar{w}_t(M_t).
\end{aligned}$$

However, the definition of M_t ensures that $\bar{w}_t(M_t) = \max_{M\in\mathcal{M}} \bar{w}_t(M)$, which implies that $\bar{w}_t(M_t) \geq \bar{w}_t(M'_t)$. This is a contradiction, and therefore we have $p_t \neq e$ for this case.

Case (2): $(e \in M_* \wedge e \in c_-) \vee (e \notin M_* \wedge e \in c_+)$**.**

First, we claim that $\tilde{M}_t \neq M_*$. Suppose that $e \in M_*$ and $e \in c_-$. Then, we see that $e \notin \tilde{M}_t$, which implies that $\tilde{M}_t \neq M_*$. On the other hand, suppose that $e \notin M_*$ and $e \in c_+$, then $e \in \tilde{M}_t$, which also implies that $\tilde{M}_t \neq M_*$. Therefore we have $\tilde{M}_t \neq M_*$ in either cases.

Hence, by Lemma 2.2, there exists an exchange set $b = (b_+, b_-) \in \mathcal{B}$ such that $e \in (b_+ \cup b_-)$, $b_- \subseteq (\tilde{M}_t \backslash M_*)$, $b_+ \subseteq (M_* \backslash \tilde{M}_t)$ and $(\tilde{M}_t \oplus b) \in \mathcal{M}$. Lemma 2.2 also indicates that $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e > 0$. We will use $\tilde{M}_t \oplus b$ as a gadget for this case. Note that the exchange set b defined here is different from the exchange set b used in Case (1).

Next, we define vectors $\boldsymbol{d} = \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t}$ and $\boldsymbol{d}_1 = \boldsymbol{\chi}_{\tilde{M}_t \oplus b} - \boldsymbol{\chi}_{M_t}$. Notice that Lemma 2.1 gives that $\boldsymbol{d}_1 = \boldsymbol{d} + \boldsymbol{\chi}_b$.

Then, we apply Lemma 2.3 by setting $M = M_t$ and $M' = \tilde{M}_t$. This shows that

$$\|\mathbf{rad}_t \circ \boldsymbol{d}\|_\infty \leq \max_{i:(\tilde{M}_t \backslash M_t) \cup (M_t \backslash \tilde{M}_t)} \mathrm{rad}_t(i) = \mathrm{rad}_t(e) < \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}, \tag{2.43}$$

where the last inequality follows from the assumption that $\mathrm{rad}_t(e) < \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}$.

Now, we bound quantity $\langle \bar{\boldsymbol{w}}_t, \boldsymbol{d}_1 \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_1| \rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle - \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle$ as follows

$$\begin{aligned}
& \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d}_1 \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_1| \rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle - \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle \\
= & \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_1| - |\boldsymbol{d}| \rangle \\
= & \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle + \left\langle \mathbf{rad}_t, \boldsymbol{d}_1^2 - \boldsymbol{d}^2 \right\rangle && (2.44) \\
= & \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle + \left\langle \mathbf{rad}_t, 2\boldsymbol{d} \circ \boldsymbol{\chi}_b + \boldsymbol{\chi}_b^2 \right\rangle && (2.45) \\
= & \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle + \left\langle \mathbf{rad}_t, \boldsymbol{\chi}_b^2 \right\rangle + 2 \langle \mathbf{rad}_t \circ \boldsymbol{d}, \boldsymbol{\chi}_b \rangle && (2.46) \\
\geq & \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle + 2 \langle \mathbf{rad}_t \circ \boldsymbol{d}, \boldsymbol{\chi}_b \rangle && (2.47) \\
\geq & \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - 2 \|\mathbf{rad}_t \circ \boldsymbol{d}\|_\infty \|\boldsymbol{\chi}_b\|_1 && (2.48) \\
> & \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{2\Delta_e}{3} && (2.49) \\
> & 0, && (2.50)
\end{aligned}$$

where Eq. (2.44) follows from the fact that $\boldsymbol{d}_1 \in \{-1, 0, 1\}^n$ and $\boldsymbol{d} \in \{-1, 0, 1\}^n$; Eq. (2.45) holds since $\boldsymbol{d}_1 = \boldsymbol{d} + \boldsymbol{\chi}_b$; Eq. (2.46) follows from Lemma 2.4; Eq. (2.47) follows from the assumption that ξ_t occurs and Lemma 2.7; Eq. (2.48) follows from Hölder's inequality; Eq. (2.49) is due to Eq. (2.43); and Eq. (2.50) follows from $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e > 0$.

Therefore, we have proven that

$$\langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle < \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d}_1 \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_1| \rangle . \tag{2.51}$$

However, we have

$$\begin{aligned}
\langle \bar{\boldsymbol{w}}_t, \boldsymbol{d} \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}| \rangle &= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t}| \right\rangle \\
&= \tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) && (2.52) \\
&\geq \tilde{w}_t(\tilde{M}_t \oplus b) - \tilde{w}_t(M_t) && (2.53) \\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_t \oplus b} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{\tilde{M}_t \oplus b} - \boldsymbol{\chi}_{M_t}| \right\rangle \\
&= \langle \bar{\boldsymbol{w}}_t, \boldsymbol{d}_1 \rangle + \langle \mathbf{rad}_t, |\boldsymbol{d}_1| \rangle , && (2.54)
\end{aligned}$$

where Eq. (2.52) follows from Lemma 2.5; and Eq. (2.53) follows from the fact that $\tilde{w}_t(\tilde{M}_t) = \max_{M \in \mathcal{M}} \tilde{w}_t(M)$. This contradicts to Eq. (2.51) and therefore $p_t \neq e$. □

2.8.4 Proof of Theorem 2.1

Theorem 2.1 is now a straightforward corollary of Lemma 2.9 and Lemma 2.10. For the reader's convenience, we first restate Theorem 2.1 in the following.

Theorem 4.2. *Given any $\delta \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\boldsymbol{w} \in \mathbb{R}^n$. Assume that the reward distribution φ_e for each arm $e \in [n]$ has mean $w(e)$ with an R-sub-Gaussian tail. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ denote the optimal set. Set $\mathrm{rad}_t(e) = R\sqrt{2 \log\left(\frac{4nt^3}{\delta}\right) / T_t(e)}$ for all $t > 0$ and $e \in [n]$. Then, with probability at least $1 - \delta$, the* CLUCB *algorithm (Algorithm 1) returns the optimal set* $\mathsf{Out} = M_*$ *and*

$$T \leq O\left(R^2 \mathit{width}(\mathcal{M})^2 \mathbf{H} \log\left(R^2 \mathbf{H}/\delta\right)\right), \tag{4.4}$$

where T denotes the number of samples used by Algorithm 1, $\mathbf{H}$ is defined in Eq. (2.2) and $\mathit{width}(\mathcal{M})$ is defined in Eq. (2.4).

Proof. Lemma 2.8 indicates that the event $\xi \triangleq \bigcap_{t=1}^{\infty} \xi_t$ occurs with probability at least $1-\delta$. In the rest of the proof, we shall assume that this event holds.

By Lemma 2.9 and the assumption on ξ, we see that $\mathsf{Out} = M_*$. Next, we focus on bounding the total number T of samples.

Fix any arm $e \in [n]$. Let $T(e)$ denote the total number of pull of arm $e \in [n]$. Let t_e be the last round which arm e is pulled, which means that $p_{t_e} = e$. It is easy to see that $T_{t_e}(e) = T(e) - 1$. By Lemma 2.10, we see that $\mathrm{rad}_{t_e}(e) \geq \frac{\Delta_e}{3\mathrm{width}(\mathcal{M})}$. Using the definition of rad_{t_e}, we have

$$\frac{\Delta_e}{3\mathrm{width}(\mathcal{M})} \leq R\sqrt{\frac{2\log\left(4nt_e^3/\delta\right)}{T(e)-1}} \leq R\sqrt{\frac{2\log\left(4nT^3/\delta\right)}{T(e)-1}}. \tag{2.55}$$

Solving Eq. (2.55) for $T(e)$, we obtain

$$T(e) \leq \frac{18\mathrm{width}(\mathcal{M})^2 R^2}{\Delta_e^2}\log(4nT^3/\delta) + 1. \tag{2.56}$$

Now we define $\tilde{\mathbf{H}} = \max\left\{\mathrm{width}(\mathcal{M})^2 R^2 \mathbf{H}, 1\right\}$. In the rest of the proof, we show that

$$T \leq 499\tilde{\mathbf{H}}\log\left(4n\tilde{\mathbf{H}}/\delta\right) + 2n. \tag{2.57}$$

Notice that the theorem follows immediately from Eq. (2.57).

If $n \geq \frac{1}{2}T$, then we see that $T \leq 2n$ and therefore Eq. (2.57) holds immediately. Next we assume that $n < \frac{1}{2}T$. Since $T > n$, we can write

$$T = C\tilde{\mathbf{H}}\log\left(4n\tilde{\mathbf{H}}/\delta\right) + n, \text{ for some } C > 0. \tag{2.58}$$

If $C \leq 499$, then it is clear that Eq. (2.57) holds. Next, we assume, in the contrary, that $C > 499$. Notice that $T = \sum_{e\in[n]} T(e)$. By summing up Eq. (2.56) for all $e \in [n]$, we have

$$T \leq n + \sum_{e\in[n]} \frac{18\mathrm{width}(\mathcal{M})^2 R^2}{\Delta_e^2}\log(4nT^3/\delta)$$

$$
\begin{aligned}
&\le n + 18\tilde{\mathbf{H}} \log(4nT^3/\delta) \\
&= n + 18\tilde{\mathbf{H}} \log(4n/\delta) + 54\tilde{\mathbf{H}} \log(T) \\
&\le n + 18\tilde{\mathbf{H}} \log(4n/\delta) + 54\tilde{\mathbf{H}} \log\left(2C\tilde{\mathbf{H}} \log\left(4n\tilde{\mathbf{H}}/\delta\right)\right) \qquad (2.59) \\
&\le n + 18\tilde{\mathbf{H}} \log(4n/\delta) + 54\tilde{\mathbf{H}} \log(2C) \\
&\qquad + 54\tilde{\mathbf{H}} \log(\tilde{\mathbf{H}}) + 54\tilde{\mathbf{H}} \log\log\left(4n\tilde{\mathbf{H}}/\delta\right) \\
&\le n + 18\tilde{\mathbf{H}} \log(4n\tilde{\mathbf{H}}/\delta) + 54\tilde{\mathbf{H}} \log(2C) \log(4n\tilde{\mathbf{H}}/\delta) \\
&\qquad + 54\tilde{\mathbf{H}} \log(4n\tilde{\mathbf{H}}/\delta) + 54\tilde{\mathbf{H}} \log(4n\tilde{\mathbf{H}}/\delta) \qquad (2.60) \\
&= n + (126 + 54\log(2C))\tilde{\mathbf{H}} \log(4n\tilde{\mathbf{H}}/\delta) \\
&< n + C\tilde{\mathbf{H}} \log(4n\tilde{\mathbf{H}}/\delta) \qquad (2.61) \\
&= T, \qquad (2.62)
\end{aligned}
$$

where Eq. (2.59) follows from Eq. (2.58) and the assumption that $n < \frac{1}{2}T$; Eq. (2.60) follows from the fact that $\tilde{\mathbf{H}} \ge 1$; Eq. (2.61) follows since $126 + 54\log(2C) < C$ for all $C > 499$; and Eq. (2.62) is due to Eq. (2.58). Hence we see that Eq. (2.62) is a contradiction. Therefore we see that $C \le 499$ which means that Eq. (2.57) holds.

□

2.9 Extensions of CLUCB

CLUCB is a general and flexible learning algorithm for the CPE problem. In this section, we present two extensions to CLUCB that allow it to work in the fixed budget setting and PAC learning setting.

2.9.1 Fixed Budget Setting

We can extend the CLUCB algorithm to the fixed budget setting using two simple modifications: (1) requiring CLUCB to terminate after T rounds; and (2) using a different construction of confidence intervals. The first modification ensures that CLUCB uses at most T samples, which meets the requirement of the fixed budget setting. And the second modification bounds the probability that the confidence intervals are valid for all arms in T rounds. The following theorem shows that the probability of error of the modified CLUCB is bounded by $O\left(Tn\exp\left(\frac{-T}{\mathrm{width}(\mathcal{M})^2\mathbf{H}}\right)\right)$.

Theorem 2.4. *Use the same notations as in Theorem 2.1. Given $T > n$ and parameter $\alpha > 0$, set the confidence radius $\mathrm{rad}_t(e) = R\sqrt{\frac{\alpha}{T_t(e)}}$ for all arms $e \in [n]$ and all $t > 0$. Run CLUCB algorithm for at most T rounds. Then, for $0 \leq \alpha \leq \frac{1}{9}(T - n)\left(R^2 width(\mathcal{M})^2\mathbf{H}\right)^{-1}$, we have*

$$\Pr\left[\mathsf{Out} \neq M_*\right] \leq 2Tn\exp\left(-\alpha/2\right). \tag{2.63}$$

In particular, the right-hand side of Eq. (2.63) *equals to $O\left(Tn\exp\left(\frac{-T}{width(\mathcal{M})^2\mathbf{H}}\right)\right)$ when parameter $\alpha = O(T\mathbf{H}^{-1}width(\mathcal{M})^{-2})$.*

Theorem 2.4 shows that the modified CLUCB algorithm in the fixed budget setting requires the knowledge of quantity $\mathbf{H}$ in order to achieve the optimal performance. However $\mathbf{H}$ is usually unknown. Therefore, although its probability of error guarantee matches the parameter-free CSAR algorithm up to logarithmic factors, this modified algorithm is considered more restricted than CSAR. Nevertheless, Theorem 2.4 shows that CLUCB can solve CPE in both fixed confidence and fixed budget settings and more importantly this theorem provides additional insights on the behavior CLUCB.

2.9.2 PAC Learning

Now we consider a setting where the learner is only required to report an approximately optimal set of arms. More specifically, we consider the notion of (ϵ,δ)-PAC algorithm. Formally, an algorithm $\mathbb{A}$ is called an (ϵ,δ)-PAC algorithm if its output $\mathsf{Out} \in \mathcal{M}$ satisfies $\Pr\left[w(M_*) - w(\mathsf{Out}) > \epsilon\right] \le \delta$.

We show that a simple modification on the CLUCB algorithm gives an (ϵ,δ)-PAC algorithm, with guarantees similar to Theorem 2.1. In fact, the only modification needed is to change the stopping condition from $\tilde{w}_t(\tilde{M}_t) = \tilde{w}_t(M_t)$ to $\tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \le \epsilon$ on line 13 of Algorithm 1. We let CLUCB-PAC denote the modified algorithm. In the following theorem, we show that CLUCB-PAC is indeed an (ϵ,δ)-PAC algorithm and has sample complexity similar to CLUCB.

Theorem 2.5. *Use the same notations as in Theorem 2.1. Fix $\delta \in (0,1)$ and $\epsilon \ge 0$. Then, with probability at least $1-\delta$, the output $\mathsf{Out} \in \mathcal{M}$ of CLUCB-PAC satisfies $w(M_*) - w(\mathsf{Out}) \le \epsilon$. In addition, the number of samples T used by the algorithm satisfies*

$$T \le O\left(R^2 \sum_{e\in[n]} \min\left\{\frac{width(\mathcal{M})^2}{\Delta_e^2}, \frac{K^2}{\epsilon^2}\right\} \log\left(\frac{R^2}{\delta} \sum_{e\in[n]} \min\left\{\frac{width(\mathcal{M})^2}{\Delta_e^2}, \frac{K^2}{\epsilon^2}\right\}\right)\right), \tag{2.64}$$

where $K = \max_{M\in\mathcal{M}} |M|$ is the size of the largest member of decision class.

We see that if $\epsilon = 0$, the sample complexity Eq. (2.64) of CLUCB-PAC equals to that of CLUCB. And the sample complexity of CLUCB-PAC decreases when ϵ increases.

There are several PAC learning algorithms dedicated for the TopK problem in the literature with different guarantees [81, 156, 61]. Zhou et al. [156] proposed an (ϵ,δ)-PAC algorithm for the TopK problem with a problem-independent sample complexity bound of $O(\frac{K^2 n}{\epsilon^2} + \frac{Kn\log(1/\delta)}{\epsilon^2})$.[2] If we ignore logarithmic

[2] We notice that the definition of Zhou et al. [156] allow an (ϵ',δ)-PAC algorithm to produce an output with *average* sub-optimality of ϵ'. This is equivalent to our definition of (ϵ,δ)-PAC algorithm with $\epsilon = K\epsilon'$ for the TopK problem. In this chapter, we translate their guarantees to our definition of PAC algorithm.

factors, then the sample complexity bound of CLUCB-PAC for the TopK problem is better than theirs since $\sum_{e\in[n]} \min\{\Delta_e^{-2}, K^2\epsilon^{-2}\} \leq nK^2\epsilon^{-2}$. On the other hand, the algorithms of Kalyanakrishnan et al. [81], Gabillon et al. [61] and Kaufmann and Kalyanakrishnan [82] guarantee to find K arms such that each of them is better than the K-th optimal arm within a factor of ϵ with probability $1-\delta$. Unless $\epsilon = 0$, their guarantee is different from ours which concerns the optimality of the sum of K arms.

2.9.3 Proof of Extension Results

Fixed Budget Setting (Theorem 2.4)

In this part, we analyze the probability of error of the modified CLUCB algorithm in the fixed budget setting and prove Theorem 2.4. First, we prove a lemma which characterizes the confidence intervals constructed in Theorem 2.4.

Lemma 2.11. *Fix parameter $\alpha > 0$ and the number of rounds $T > 0$. Assume that the reward distribution φ_e is a R-sub-Gaussian distribution for all $e \in [n]$. Let the confidence radius $\mathrm{rad}_t(e)$ of arm $e \in [n]$ and round $t > 0$ be $\mathrm{rad}_t(e) = R\sqrt{\frac{\alpha}{T_t(e)}}$. Then, we have*

$$\Pr\left[\bigcap_{t=1}^{T} \xi_t\right] \geq 1 - 2nT\exp\left(-\alpha/2\right),$$

where ξ_t is the random event defined in Eq. (2.12).

Proof. For any $t > 0$ and $e \in [n]$, using Hoeffding's inequality, we have

$$\Pr\left[\left|\bar{w}_t(e) - w(e)\right| \geq \mathrm{rad}_t(e)\right] \leq 2\exp(-\alpha/2).$$

By a union bound over all arms $e \in [n]$, we see that $\Pr[\xi_t] \geq 1 - 2n\exp(-\alpha/2)$. The lemma follows immediately by using union bound again over all round $t \in [T]$. □

Theorem 2.4 can be obtained from the key lemmas (Lemma 2.9 and Lemma 2.10) and Lemma 2.11.

Proof of Theorem 2.4. Define random event $\xi = \bigcap_{t=1}^{T} \xi_t$. By Lemma 2.11, we see that $\Pr[\xi] \geq 1 - 2nT\exp(-\alpha/2)$. In the rest of the proof, we assume that ξ happens.

Let T^* denote the round that the algorithm stops. We claim that the algorithm stops before the budget is exhausted, i.e., $T^* < T$. If the claim is true, then the algorithm stops since it meets the stopping condition on round T^*. Hence $\tilde{w}_t(\tilde{M}_{T^*}) = \tilde{w}_t(M_{T^*})$ and $\mathsf{Out} = M_{T^*}$. By assumption on ξ and Lemma 2.9, we know that $M_{T^*} = M_*$. Therefore the theorem follows immediately from this claim and the bound of $\Pr[\xi]$.

Next, we show that this claim is true. Let $T(e)$ denote the total number of pulls of arm $e \in [n]$. Let t_e be the last round that arm e is pulled. Hence $T_{t_e}(e) = T_e - 1$. By Lemma 2.10, we see that $\mathrm{rad}_{t_e}(e) \geq \frac{\Delta}{3\mathrm{width}(\mathcal{B})}$. Now plugging in the definition of $\mathrm{rad}_{t_e}(e)$, we have

$$\begin{aligned} \frac{\Delta}{3\mathrm{width}(\mathcal{B})} &\leq \mathrm{rad}_{t_e}(e) \\ &= R\sqrt{\frac{\alpha}{T_{t_e}(e)}} = R\sqrt{\frac{\alpha}{T(e) - 1}}. \end{aligned}$$

Hence we have

$$T_e \leq \frac{9R^2\mathrm{width}(\mathcal{B})^2}{\Delta_e^2} \cdot \alpha + 1. \tag{2.65}$$

By summing up Eq. (2.65) for all $e \in [n]$, we have

$$T^* = \sum_{e \in [n]} T_e \leq \alpha \cdot 9R^2\mathrm{width}(\mathcal{B})^2 \left(\sum_{e \in [n]} \Delta_e^{-2} \right) + n < T,$$

where we have used the assumption that $\alpha < \frac{1}{9}(T-n) \cdot \left(R^2\mathrm{width}(\mathcal{B})^2 \left(\sum_{e \in [n]} \Delta_e^{-2} \right) \right)$

□

PAC Learning (Theorem 2.5)

First, we prove a (ϵ, δ)-PAC counterpart of Lemma 2.9.

Lemma 2.12. *If* CLUCB-PAC *stops on round t and suppose that event ξ_t occurs. Then, we have $w(M_*) - w(\mathsf{Out}) \le \epsilon$.*

Proof. By definition, we know that $\mathsf{Out} = M_t$. Notice that the stopping condition of CLUCB-PAC ensures that $\tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \le \epsilon$. Therefore, we have

$$\begin{aligned}
\epsilon &\ge \tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \\
&\ge \tilde{w}_t(M_*) - \tilde{w}_t(M_t) && (2.66)\\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_*} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{M_*} - \boldsymbol{\chi}_{M_t}| \right\rangle && (2.67)\\
&\ge \left\langle \boldsymbol{w}, \boldsymbol{\chi}_{M_*} - \boldsymbol{\chi}_{M_t} \right\rangle && (2.68)\\
&= w(M_*) - w(M_t),
\end{aligned}$$

where Eq. (2.66) follows from the definition that $\tilde{w}_t(\tilde{M}_t) = \max_{M \in \mathcal{M}} \tilde{w}_t(M)$; Eq. (2.67) follows from Lemma 2.5; Eq. (2.68) follows from the assumption that ξ_t occurs and Lemma 2.7. □

The next lemma generalizes Lemma 2.10. It shows that on event ξ_t each arm $e \in [n]$ will not be played on round t if $\mathrm{rad}_t(e) < \max\left\{\frac{\Delta_e}{3\mathrm{width}(\mathcal{M})}, \frac{\epsilon}{2K}\right\}$.

Lemma 2.13. *Let $K = \max_{M \in \mathcal{M}} |M|$. For any arm $e \in [n]$ and any round $t > n$ after initialization, if $\mathrm{rad}_t(e) < \max\left\{\frac{\Delta_e}{3width(\mathcal{M})}, \frac{\epsilon}{2K}\right\}$ and random event ξ_t occurs, then arm e will not be played on round t, i.e., $p_t \ne e$.*

Proof. If $\mathrm{rad}_t(e) < \frac{\Delta_e}{3\mathrm{width}(\mathcal{M})}$, then we can apply Lemma 2.10 which immediately gives that $p_t \ne e$. Hence, we only need to prove the case that $\frac{\Delta_e}{3\mathrm{width}(\mathcal{M})} \le \mathrm{rad}_t(e) < \frac{\epsilon}{2K}$.

Now suppose that $p_t = e$. By the choice of p_t, we know that for each $i \in (M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)$, we have $\mathrm{rad}_t(i) \le \mathrm{rad}_t(e) < \frac{\epsilon}{2K}$.

By summing up this inequality for all $i \in (M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)$, we have

$$\begin{aligned} \epsilon &> \sum_{i \in (M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)} \mathrm{rad}_t(i) && (2.69) \\ &= \left\langle \mathbf{rad}_t, \left| \boldsymbol{\chi}_{M_t} - \boldsymbol{\chi}_{\tilde{M}_t} \right| \right\rangle, && (2.70) \end{aligned}$$

where Eq. (2.69) follows from the fact that $|(M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)| \leq |M_t| + |\tilde{M}_t| \leq 2K$; and Eq. (2.70) uses the fact that $\boldsymbol{\chi}_{(M_t \backslash \tilde{M}_t) \cup (\tilde{M}_t \backslash M_t)} = |\boldsymbol{\chi}_{M_t} - \boldsymbol{\chi}_{\tilde{M}_t}|$.

Then, we have

$$\begin{aligned} \tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) &= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t} \right\rangle + \left\langle \mathbf{rad}_t, |\boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t}| \right\rangle && (2.71) \\ &\leq \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_t} - \boldsymbol{\chi}_{M_t} \right\rangle + \epsilon && (2.72) \\ &= \bar{w}_t(\tilde{M}_t) - \bar{w}_t(M_t) + \epsilon \\ &\leq \epsilon, && (2.73) \end{aligned}$$

where Eq. (2.71) follows from Lemma 2.5; Eq. (2.72) uses Eq. (2.70); and Eq. (2.73) follows from $\bar{w}_t(M_t) \geq \bar{w}_t(\tilde{M}_t)$.

Therefore, we see that $\tilde{w}_t(\tilde{M}_t) - \tilde{w}_t(M_t) \leq \epsilon$. By the stopping condition of CLUCB-PAC, the algorithm must terminate on round t, before playing any arms. This contradicts to the assumption that $p_t = e$. □

Using Lemma 2.13 and Lemma 2.12, we are ready to prove Theorem 2.5.

Proof of Theorem 2.5. Similar to the proof of Theorem 2.1, we appeal to Lemma 2.8, which shows that the event $\xi \triangleq \bigcap_{t=1}^{\infty} \xi_t$ occurs with probability at least $1 - \delta$. And we shall assume that ξ occurs in the rest of the proof.

By the assumption of ξ and Lemma 2.12, we know that $w(M_*) - w(\mathsf{Out}) \leq \epsilon$. Therefore, we only remain to bound the number of samples T.

Consider an arbitrary arm $e \in [n]$. Let $T(e)$ denote the total number of pull of arm $e \in [n]$. Let t_e be the last round which arm e is pulled, i.e., $p_{t_e} = e$. Hence $T_{t_e}(e) = T(e) - 1$. By Lemma 2.13, we see that $\mathrm{rad}_{t_e}(e) \geq \max\{\frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}, \frac{\epsilon}{2K}\}$. Then, by the construction of $\mathrm{rad}_{t_e}(e)$, we have

$$\max\left\{\frac{\Delta_e}{3\mathrm{width}(\mathcal{B})}, \frac{\epsilon}{2K}\right\} \leq R\sqrt{\frac{2\log\left(4nt_e^3/\delta\right)}{T(e)-1}} \leq R\sqrt{\frac{2\log\left(4nT^3/\delta\right)}{T(e)-1}}. \tag{2.74}$$

Solving Eq. (2.74) for $T(e)$, we obtain

$$T(e) \leq R^2 \min\left\{\frac{18\mathrm{width}(\mathcal{B})^2}{\Delta_e^2}, \frac{16K^2}{\epsilon^2}\right\}\log(4nT^3/\delta) + 1. \tag{2.75}$$

Notice that $T = \sum_{i\in[n]} T(e)$. Hence the theorem follows by summing up Eq. (2.75) for all $e \in [n]$ and solving for T. □

2.10 Proof of Lower Bound (Theorem 2.2)

In this section, we prove the problem-dependent lower bound of the general CPE problem (Theorem 2.2). In addition, we provide evidence on the conjecture that the sample complexity should hinge on the size of exchange sets (Theorem 2.6), which is relevant for decision classes with non-constant widths.

Notations. In this section, we will use the notion of "next-to-optimal set" defined as follows. Fix a decision class $\mathcal{M} \subseteq 2^{[n]}$ and an expected reward vector $\boldsymbol{w} \in \mathbb{R}^n$. Let $M_* = \arg\max_{M\in\mathcal{M}} w(M)$ denote the optimal set. Then, for any $e \in [n]$, we define the next-to-optimal set associated with e as follows

$$M_e = \begin{cases} \arg\max_{M\in\mathcal{M}: e\in M} w(M) & \text{if } e \notin M_*, \\ \arg\max_{M\in\mathcal{M}: e\notin M} w(M) & \text{if } e \in M_*. \end{cases} \tag{2.76}$$

We note that, by definition of Δ_e in Eq. (2.1), we have $w(M_*) - w(M_e) = \Delta_e$.

2.10.1 Proof of Theorem 2.2

For reader's convenience, we restate Theorem 2.2 in the following.

Theorem 2.2. *Fix any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any vector $\boldsymbol{w} \in \mathbb{R}^n$. Suppose that, for each arm $e \in [n]$, the reward distribution φ_e is given by $\varphi_e = \mathcal{N}(w(e), 1)$, where we let $\mathcal{N}(\mu, \sigma^2)$ denote Gaussian distribution with mean μ and variance σ^2. Then, for any $\delta \in (0, e^{-16}/4)$ and any δ-correct algorithm $\mathbb{A}$, we have*

$$\mathbb{E}[T] \geq \frac{1}{16}\mathbf{H}\log\left(\frac{1}{4\delta}\right), \tag{2.6}$$

where T denote the number of total samples used by algorithm $\mathbb{A}$ and $\mathbf{H}$ is defined in Eq. (2.2)*.*

Before stating our proof, we first introduce two technical lemmas. The first lemma is the well-known Kolmogrov's inequality.

Lemma 2.14. (Kolmogrov's inequality [127, Corollary 7.66]) *Let $Z_1, \ldots, Z_n$ be independent zero-mean random variables with* $\mathrm{Var}[Z_k] \leq +\infty$ *for all $k \in [n]$. Then, for any $\lambda > 0$,*

$$\Pr\left[\max_{1 \leq k \leq n} |S_k| \geq \lambda\right] \leq \frac{1}{\lambda^2}\sum_{i=1}^{n}\mathrm{Var}[Z_k],$$

where $S_k = X_1 + \ldots + X_k$.

The second technical lemma shows that the joint likelihood of Gaussian distributions on a sequence of variables does not change much when the mean of the distribution shifts by a sufficiently small value.

Lemma 2.15. *Fix some $d \in \mathbb{R}$ and $\theta \in (0,1)$. Define $t = \frac{1}{4d^2}\log(1/\theta)$. Given any integer $T \leq 4t$ and any sequence $s_1, \ldots, s_T$.*

Let $X_1, \ldots, X_T$ be T real numbers which satisfy the following

$$\left|\sum_{i=1}^{T} X_i - \sum_{i=1}^{T} s_i\right| \leq \sqrt{t \log(1/\theta)}. \tag{2.77}$$

Then, we have

$$\prod_{i=1}^{T} \frac{\mathcal{N}(X_i|s_i + d, 1)}{\mathcal{N}(X_i|s_i, 1)} \geq \theta,$$

where we let $\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ denote the probability density function of normal distribution with mean μ and variance σ^2.

Proof. We define $v_i = X_i - s_i$ for all $i \in [T]$. Then, we have

$$\begin{aligned}
\prod_{i=1}^{T} \frac{\mathcal{N}(X_i|s_i + d, 1)}{\mathcal{N}(X_i|s_i, 1)} &= \prod_{i=1}^{T} \exp\left(\frac{-\left(X_i - s_i - d\right)^2 + (X_i - s_i)^2}{2}\right) \\
&= \prod_{i=1}^{T} \exp\left(-v_i d - \frac{1}{2} d^2\right) \\
&= \exp\left(-\sum_{i=1}^{T} v_i d\right) \exp\left(-\frac{Td^2}{2}\right). \qquad (2.78)
\end{aligned}$$

We now bound each term on the right-hand side of Eq. (2.78) as follows

$$\begin{aligned}
\exp\left(-\sum_{i=1}^{T} v_i d\right) &\geq \exp\left(-\left|\sum_{i=1}^{T} v_i\right| \cdot |d|\right) \\
&\geq \exp\left(-\sqrt{t\log(1/\theta)}d\right) \qquad (2.79) \\
&= \exp\left(-\frac{1}{2}\log(1/\theta)\right) = \theta^{1/2}, \qquad (2.80)
\end{aligned}$$

where Eq. (2.79) follows from Eq. (2.77); and Eq. (2.80) follows from the fact $t \leq \frac{1}{4d^2}\log(1/\theta)$. Next we have

$$\exp\left(-\frac{Td^2}{2}\right) \geq \exp\left(-2td^2\right) \tag{2.81}$$

$$= \exp\left(-\frac{1}{2}\log(1/\theta)\right) = \theta^{1/2}, \tag{2.82}$$

where Eq. (2.81) follows from $T \le 4t$ and Eq. (2.82) follows from the definition of t. The lemma follows immediate by combining Eq. (2.78), Eq. (2.80) and Eq. (2.82). □

Proof of Theorem 2.2. Fix $\delta > 0$, $\boldsymbol{w} = \big(w(1), \dots, w(n)\big)^T$ and a δ-correct algorithm $\mathbb{A}$. For each $e \in [n]$, assume that the reward distribution is given by $\varphi_e = \mathcal{N}(w(e), 1)$. For any $e \in [n]$, let T_e denote the number of trials of arm e used by algorithm $\mathbb{A}$. In the rest of the proof, we will show that for any $e \in [n]$, the number of trials of arm e is lower-bounded by

$$\mathbb{E}[T_e] \ge \frac{1}{16\Delta_e^2}\log(1/4\delta). \tag{2.83}$$

Notice that the theorem follows immediately by summing up Eq. (2.83) for all $e \in [n]$.

Now fix an arm $e \in [n]$. We define $\theta = 4\delta$ and $t_e^* = \frac{1}{16\Delta_e^2}\log(1/\theta)$. We prove Eq. (2.83) by contradiction. Therefore we assume the opposite that $\mathbb{E}[T_e] < t_e^*$ in the rest of the proof.

Step (1): An alternative hypothesis. We consider two hypothesis H_0 and H_1. Under hypothesis H_0, all reward distributions are same with our assumption in the theorem as follows

$$H_0 : \varphi_l = \mathcal{N}(w(l), 1) \quad \text{for all } l \in [n].$$

On the other hand, under hypothesis H_1, we change the means of reward distributions such that

$$H_1 : \varphi_e = \begin{cases} \mathcal{N}(w(e) - 2\Delta_e, 1) & \text{if } e \in M_* \\ \mathcal{N}(w(e) + 2\Delta_e, 1) & \text{if } e \notin M_* \end{cases} \quad \text{and } \varphi_l = \mathcal{N}(w(l), 1) \quad \text{for all } l \ne e.$$

For $l \in \{0, 1\}$, we use $\mathbb{E}_l$ and $\Pr_l$ to denote the expectation and probability, respectively, under the hypothesis H_l.

Now we claim that M_* is no longer the optimal set under hypothesis H_1. Let M_e denote the next-to-optimal set defined Eq. (2.76). By definition of Δ_e in Eq. (2.1), we know that $w(M_*)-w(M_e)=\Delta_e$. Let $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ be expected reward vectors under H_0 and H_1 respectively. We have

$$\begin{aligned} w_1(M_*)-w_1(M_e) &= w(M_*)-w(M_e)-2\Delta_e \\ &= -\Delta_e < 0. \end{aligned}$$

This means that under H_1, the set M_* is not the optimal set.

Step (2): Three random events. Let $X_1,\ldots,X_{T_e}$ denote the sequence of reward outcomes of arm e. Now we define three random events $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ as follows

$$\mathcal{A}=\{T_e\le 4t_e^*\},\ \mathcal{B}=\{\mathsf{Out}=M_*\} \text{ and}$$

$$\mathcal{C}=\left\{\max_{1\le t\le 4t_e^*}\left|\sum_{i=1}^{t}X_t-t\cdot w(e)\right|<\sqrt{t_e^*\log(1/\theta)}\right\},$$

where Out is the output of algorithm $\mathbb{A}$.

Now we bound the probability of these events under hypothesis H_0. First, we show that $\Pr_0[\mathcal{A}]\ge 3/4$. This can be proven by Markov's inequality as follows.

$$\Pr_0[T_e>4t_e^*]\le\frac{\mathbb{E}_0[T_e]}{4t_e^*}\le\frac{t_e^*}{4t_e^*}=\frac{1}{4}.$$

We now show that $\Pr_0[\mathcal{C}]\ge 3/4$. Notice that $\big\{X_t-w(e)\big\}_{t=1,\ldots,}$ is a sequence of zero-mean independent random variables under H_0. Define $K_t=\sum_{i=1}^t X_t$. Then, by Kolmogorov's inequality (Lemma 2.14), we have

$$\begin{aligned}\Pr_0\left[\max_{1\le t\le 4t_e^*}|K_t-t\cdot w(e)|\ge\sqrt{t_e^*\log(1/\theta)}\right] &\le \frac{\mathbb{E}_0[(K_{4t_e^*}-4w(e)t_e^*)^2]}{t_e^*\log(1/\theta)} \\ &\overset{(a)}{=}\frac{4t_e^*}{t_e^*\log(1/\theta)}\overset{(b)}{<}\frac{1}{4},\end{aligned}$$

where (a) follows from the fact that the variance of φ_e equals to 1 and therefore $\mathbb{E}_0[(K_{4t_e^*} - 4w(e)t_e^*)^2] = 4t_e^*$; and (b) follows since $\theta < e^{-16}$.

Since the probability of error of algorithm $\mathbb{A}$ is at most $\delta < e^{-16}/4 < 1/4$, we have $\Pr_0[\mathcal{B}] \geq 3/4$. Define random event $\mathcal{S} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$. Then, by union bound, we have $\Pr_0[\mathcal{S}] \geq 1/4$.

Step (3): The loss of likelihood. Now, we claim that, under the assumption that $\mathbb{E}_0[T_e] < t_e^*$, one has $\Pr_1[\mathcal{B}] \geq \delta$. Let W be the history of the sampling process until the algorithm stops (including the sequence of arms chosen at each time and the sequence of observed outcomes). Define the likelihood function L_l as

$$L_l(w) = p_l(W = w),$$

where p_l is the probability density function under hypothesis H_l.

Now assume that the event $\mathcal{S}$ occurred. We will bound the likelihood ratio $L_1(W)/L_0(W)$ under this assumption. Since H_1 and H_0 only differs on the reward distribution of arm e, we have

$$\frac{L_1(W)}{L_0(W)} = \prod_{i=1}^{T_e} \frac{\mathcal{N}(X_i|w_1(e), 1)}{\mathcal{N}(X_i|w_0(e), 1)}. \tag{2.84}$$

By definition of H_1 and H_0, we see that $w_1(e) = w_0(e) \pm 2\Delta_e$ (where the sign depends on whether $e \in M_*$). Therefore, when event $\mathcal{S}$ occurs, it easy to verify that we can apply Lemma 2.15 (by setting $d = w_1(e) - w_0(e) = \pm 2\Delta_e$, $T = T_e$ and $s_i = w_0(e)$ for all i). Hence, by Lemma 2.15 and Eq. (2.84), we have

$$\frac{L_1(W)}{L_0(W)} \geq \theta = 4\delta$$

holds if event $\mathcal{S}$ occurs.

Then, define 1_S as the indicator variable of event $\mathcal{S}$, i.e., $1_S = 1$ if and only if $\mathcal{S}$ occurs and otherwise $1_S = 0$. Then, we have

$$\frac{L_1(W)}{L_0(W)} 1_S \geq 4\delta 1_S$$

holds regardless the occurrence of event $\mathcal{S}$. Therefore, we can obtain

$$\begin{aligned}\Pr_1[\mathcal{B}] &\geq \Pr_1[\mathcal{S}] = \mathbb{E}_1[1_S] \\ &= \mathbb{E}_0\left[\frac{L_1(W)}{L_0(W)} 1_S\right] \\ &\geq 4\delta \mathbb{E}_0[1_S] \\ &= 4\delta \Pr_0[\mathcal{S}] \geq \delta.\end{aligned}$$

Now we have proven that, if $\mathbb{E}_0[T_e] < t_e^*$, then $\Pr_1[\mathcal{B}] \geq \delta$. This means that, if $\mathbb{E}_0[T_e] < t_e^*$, algorithm $\mathbb{A}$ will choose M_* as the output with probability at least δ, under hypothesis H_1. However, under H_1, we have shown that M_* is not the optimal set since $w_1(M_e) > w_1(M_*)$. Therefore, algorithm $\mathbb{A}$ has a probability of error at least δ under H_1. This contradicts to the assumption that algorithm $\mathbb{A}$ is a δ-correct algorithm. Hence, we must have $\mathbb{E}_0[T_e] \geq t_e^* = \frac{1}{16\Delta_e^2} \log(1/4\delta)$. □

2.10.2 Exchange Set Size Dependent Lower Bound

As a supplement to our main lower bound (Theorem 2.2), we show that, for any arm $e \in [n]$, there exists an exchange set $b = (b_+, b_-)$ which contains e such that a δ-correct algorithm must spend $\tilde{\Omega}\Big(\big(|b_+|+|b_-|\big)^2/\Delta_e^2\Big)$ samples on exploring the arms belonging to $b_+ \cup b_-$. Hence, on average, each arm $e \in b_+ \cup b_-$ must be sampled for $\tilde{\Omega}((|b_+| + |b_-|)\Delta_e^{-2})$ times. This is asymptotically stronger than the result of Theorem 2.2 when the size of corresponding exchange set $|b_+| + |b_-|$ is non-constant. This result is formalized in the following theorem.

Theorem 2.6. *Fix any $\mathcal{M} \subseteq 2^{[n]}$ and any vector $\boldsymbol{w} \in \mathbb{R}^n$. Suppose that, for each arm $e \in [n]$, the reward distribution φ_e is given by $\varphi_e = \mathcal{N}(w(e), 1)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2. Fix any $\delta \in (0, e^{-16}/4)$ and any δ-correct algorithm $\mathbb{A}$.*

Then, for any $e \in [n]$, there exists an exchange set $b = (b_+, b_-)$, such that $e \in b_+ \cup b_-$ and

$$\mathbb{E}\left[\sum_{i \in b_+ \cup b_-} T_i\right] \geq \frac{(|b_+| + |b_-|)^2}{32\Delta_e^2} \log(1/4\delta),$$

where T_i is the number of samples of arm i.

The proof is quite similar to that of Theorem 2.2 except that they use different constructions of alternative hypothesis and consequently this introduces some difference on the details of computations.

Proof. Fix $\delta > 0$, $\boldsymbol{w} = \big(w(1), \ldots, w(n)\big)^T$ and a δ-correct algorithm $\mathbb{A}$. For each $i \in [n]$, assume that the reward distribution is given by $\varphi_i = \mathcal{N}(w(i), 1)$. For any $i \in [n]$, let T_i denote the number of trials of arm i used by algorithm $\mathbb{A}$.

Step (0): Setup. Fix an arm $e \in [n]$. As the first step, we construct the exchange set $b = (b_+, b_-)$ claimed in the theorem. Let M_e denote the next-to-optimal set as defined in Eq. (2.76). By definition of Δ_e in Eq. (2.1), we know that $w(M_*) - w(M_e) = \Delta_e$. We construct the exchange set $b = (b_+, b_-)$ where $b_+ = M_* \backslash M_e$ and $b_- = M_e \backslash M_*$. It is easy to check that $M_e \oplus b = M_*$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle = \Delta_e > 0$.

We have now constructed the exchange set. We define $T_{b_-} = \sum_{i \in b_-} T_i$ and $T_{b_+} = \sum_{i \in b_+} T_i$. Now we claim that

$$\textbf{(a)}\;\; \mathbb{E}\left[T_{b_-}\right] \geq \frac{|b_-|^2}{16\Delta_e^2} \log(1/4\delta) \quad \text{and} \quad \textbf{(b)}\;\; \mathbb{E}\left[T_{b_+}\right] \geq \frac{|b_+|^2}{16\Delta_e^2} \log(1/4\delta). \tag{2.85}$$

It is easy to check that theorem follows immediately from claims **(a)** and **(b)**. In the rest of the proof, we focus on claim **(a)**; the claim **(b)** can be proven using an almost identical similar argument.

Now we define $\theta = 4\delta$ and $t^*_{b_-} = \frac{|b_-|^2}{16\Delta_e^2}\log(1/\theta)$. We prove claim **(a)** by contradiction, that is to assume the opposite that $\mathbb{E}[T_{b_-}] < t^*_{b_-}$.

Step (1): An alternative hypothesis. We define two hypotheses H_0 and H_1. Under hypothesis H_0, the reward distribution

$$H_0 : \varphi_l = \mathcal{N}(w(l), 1) \quad \text{for all } l \in [n].$$

Under hypothesis H_1, the mean reward of each arm is given by

$$H_1 : \varphi_i = \begin{cases} \mathcal{N}\left(w(i) + \frac{2\Delta_e}{|b_-|}, 1\right) & \text{if } i \in b_-, \\ \mathcal{N}(w(i), 1) & \text{if } i \notin b_-. \end{cases}$$

Similar to the proof of Theorem 2.2, we let $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ denote the expected reward vectors under H_0 and H_1 respectively. One can verify that $w_1(M_*) - w_1(M_e) = -\Delta_e < 0$. This means that under H_1, the set M_* is not the optimal set.

Step (2): Three random events. First we consider the complete sequence of sampling process by algorithm $\mathbb{A}$. Formally, let $W = \{(\tilde{I}_1, \tilde{X}_1), \ldots, (\tilde{I}_T, \tilde{X}_T)\}$ be the sequence of all trials by algorithm $\mathbb{A}$, where $\tilde{I}_i$ denotes the arm played in i-th trial and $\tilde{X}_i$ be the reward outcome of i-th trial. Then, consider the subsequence W_1 of W which consists of all the trials of arms in b_-. Specifically, we write $W = \{(I_1, X_1), \ldots, (I_{T_{b_-}}, X_{T_{b_-}})\}$ such that W_1 is a subsequence of W and $I_i \in b_-$ for all i.

Now we define three random events $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ as follows

$$\mathcal{A} = \{T_{b_-} \le 4t^*_{b_-}\},\ \mathcal{B} = \{\mathsf{Out} = M_*\} \text{ and}$$

$$\mathcal{C} = \left\{ \max_{1 \le t \le 4t^*_{b_-}} \left| \sum_{i=1}^{t} X_i - \sum_{i=1}^{t} w(I_i) \right| < \sqrt{t^*_{b_-} \log(1/\theta)} \right\},$$

where Out is the output of algorithm $\mathbb{A}$. We now bound the probability of each event. First, by Markov's inequality, we have

$$\Pr_0[T_{b_-} > 4t^*_{b_-}] \le \frac{\mathbb{E}_0[T_{b_-}]}{4t^*_{b_-}} = \frac{t^*_{b_-}}{4t^*_{b_-}} = \frac{1}{4}.$$

Next, using Kolmogrov's inequality (Lemma 2.14), we obtain

$$\begin{aligned}
&\Pr_0\left[\max_{1\le t\le 4t^*_{b_-}}\left|\sum_{i=1}^{t} X_i - \sum_{i=1}^{t} w(I_i)\right| \ge \sqrt{t^*_e \log(1/\theta)}\right]\\
\le&\frac{\mathbb{E}_0\left[\left(\sum_{i=1}^{4t^*_{b_-}} X_i - \sum_{i=1}^{4t^*_{b_-}} w(I_i)\right)^2\right]}{t^*_e \log(1/\theta)}\\
\overset{(a)}{=}&\frac{4t^*_{b_-}}{t^*_{b_-}\log(1/\theta)} \overset{(b)}{<} \frac{1}{4},
\end{aligned}$$

where (a) follows from the fact that all reward distributions have unit variance; and (b) follows since $\theta < e^{-16}$.

Since $\mathbb{A}$ is δ-correct algorithm and $\delta < 1/4$, we have $\Pr_0[\mathcal{B}] \ge 3/4$. Therefore, we have that the random event $\mathcal{S} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$ occurs with probability at least $1/4$ under H_0.

Step (3): The loss of likelihood. Similar to the proof of Theorem 2.2, we let L_l denote the likelihood function under hypothesis H_l for $l \in \{0, 1\}$. Since the difference of H_0 and H_1 only lies in the reward distributions of arms belonging to b_-, we have

$$\frac{L_1(W)}{L_0(W)} = \prod_{i=1}^{T_{b_-}} \frac{\mathcal{N}(X_i|w_1(I_i), 1)}{\mathcal{N}(X_i|w_0(I_i), 1)},$$

where X_i and I_i is as defined in Step (2). Assume that $\mathcal{S}$ occurs. Since, for all $i \in [T_{b_-}]$, we have $w_1(I_i) - w_0(I_i) = \frac{2\Delta_e}{|b_-|}$, we can apply Lemma 2.15 here (by setting $d = \frac{2\Delta_e}{|b_-|}$). Therefore, on event $\mathcal{S}$, we have

$$\frac{L_1(W)}{L_0(W)} \ge \theta.$$

The rest of the proof is identical to Step (3) in the proof of Theorem 2.2, and one can show that $\Pr_1[\mathcal{B}] \ge \delta$ under the assumption that $\mathbb{E}[T_{b_-}] < t^*_{b_-}$. This means the probability of error of algorithm $\mathbb{A}$ is at least δ. This contradicts to the assumption of $\mathbb{A}$. Therefore we have $\mathbb{E}[T_{b\_}] \geq t^*_{b\_}$ which proves claim **(a)** in Eq. (2.85). □

2.11 Analysis of CSAR (Theorem 2.3)

Notations. Let $\boldsymbol{w} \in \mathbb{R}^n$ be the vector of the expected rewards of arms. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ be the optimal set. Let T be the budget of samples. We will also use the following additional notations in the rest of this section. Let $M \subseteq [n]$ be a set, we denote $\neg M$ to be the complement of M. Let $\Delta_{(1)}, \ldots, \Delta_{(n)}$ be a permutation of $\Delta_1, \ldots, \Delta_n$ such that $\Delta_{(1)} \leq \ldots\ldots \Delta_{(n)}$. Let $A_1, \ldots, A_n$ and $B_1, \ldots, B_n$ be the two sequences of sets which are defined in Algorithm 2. We will also continue to use the notations of incidence vectors of sets and exchange sets, which are defined in Section 2.8.

2.11.1 Confidence Intervals

First we establish the confidence bounds used for the analysis of CSAR.

Lemma 2.16. *Given a phase $t \in [n]$, we define random event τ_t as follows*

$$\tau_t = \left\{ \forall i \in [n] \backslash (A_t \cup B_t) \quad \left|\bar{w}_t(i) - w(i)\right| < \frac{\Delta_{(n-t+1)}}{3width(\mathcal{M})} \right\}. \tag{2.86}$$

Then, we have

$$\Pr\left[\bigcap_{t=1}^{n} \tau_t\right] \geq 1 - n^2 \exp\left(-\frac{(T-n)}{18R^2 \tilde{\log}(n)\, width(\mathcal{M})^2 \mathbf{H}_2}\right). \tag{2.87}$$

Proof. Fix some $t \in [n]$ and fix some active arm $i \in [n]\backslash(A_t \cup B_t)$ of phase t.

Notice that the arm i has been pulled for $\tilde{T}_t$ times during the first t phases. Therefore, by Hoeffding's inequality (Lemma 2.6), we have

$$\Pr\left[\left|\bar{w}_t(i) - w(i)\right| \geq \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{M})}\right] \leq 2\exp\left(-\frac{\tilde{T}_t\Delta^2_{(n-t+1)}}{18R^2\mathrm{width}(\mathcal{M})^2}\right). \tag{2.88}$$

By plugging the definition of $\tilde{T}_t$, the quantity $\tilde{T}_t\Delta^2_{(n-t+1)}$ on the right-hand side of Eq. (2.88) can be further bounded by

$$\begin{aligned}\tilde{T}_t\Delta^2_{(n-t+1)} &\geq \frac{T-n}{\tilde{\log}(n)(n-t+1)}\Delta^2_{(n-t+1)}\\ &\geq \frac{T-n}{\tilde{\log}(n)\mathbf{H}_2},\end{aligned}$$

where the last inequality follows from the definition of $\mathbf{H}_2 = \max_{i\in n} i\Delta^{-2}_{(i)}$. By plugging the last inequality into Eq. (2.88), we have

$$\Pr\left[\left|\bar{w}_t(i) - w(i)\right| \geq \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{M})}\right] \leq 2\exp\left(-\frac{(T-n)}{18R^2\tilde{\log}(n)\mathrm{width}(\mathcal{M})^2\mathbf{H}_2}\right). \tag{2.89}$$

Now using Eq. (2.89) and a union bound for all $t \in [n]$ and all $i \in [n]\backslash(A_t \cup B_t)$, we have

$$\begin{aligned}\Pr\left[\bigcap_{t=1}^{n}\tau_t\right] &\geq 1 - 2\sum_{t=1}^{n}(n-t+1)\exp\left(-\frac{(T-n)}{18R^2\tilde{\log}(n)\mathrm{width}(\mathcal{M})^2\mathbf{H}_2}\right)\\ &\geq 1 - n^2\exp\left(-\frac{(T-n)}{18R^2\tilde{\log}(n)\mathrm{width}(\mathcal{M})^2\mathbf{H}_2}\right).\end{aligned}$$

□

Readers may notice that the right-hand side of Eq. (2.87) equals to the probability of error of CSAR claimed in Theorem 2.3. Indeed, we will show that the CSAR algorithm will not make any mistakes if the random event $\bigcap_{t=1}^{n}\tau_t$ occurs.

The following lemma builds the confidence bound of inner products.

Lemma 2.17. *Fix a phase $t \in [n]$, suppose that random event τ_t occurs. For any vector $\boldsymbol{a} \in \mathbb{R}^n$, suppose that $\operatorname{supp}(\boldsymbol{a}) \cap (A_t \cup B_t) = \emptyset$, where $\operatorname{supp}(\boldsymbol{a}) \triangleq \{i \mid a(i) \neq 0\}$ is the support of vector $\boldsymbol{a}$. Then, we have*

$$|\langle \bar{\boldsymbol{w}}_t, \boldsymbol{a} \rangle - \langle \boldsymbol{w}, \boldsymbol{a} \rangle| < \frac{\Delta_{(n-t+1)}}{3width(\mathcal{M})} \|\boldsymbol{a}\|_1 .$$

Proof. Suppose that τ_t occurs. Then, similar to the proof of Lemma 2.7, we have

$$\begin{aligned}
|\langle \bar{\boldsymbol{w}}_t, \boldsymbol{a} \rangle - \langle \boldsymbol{w}, \boldsymbol{a} \rangle| &= |\langle \bar{\boldsymbol{w}}_t - \boldsymbol{w}, \boldsymbol{a} \rangle| \\
&= \left| \sum_{i=1}^{n} \big(\bar{w}_t(i) - w(i)\big) a(i) \right| \\
&\leq \left| \sum_{i \in [n] \setminus (A_t \cup B_t)} \big(\bar{w}_t(i) - w(i)\big) a(i) \right| \qquad (2.90) \\
&\leq \sum_{i \in [n] \setminus (A_t \cup B_t)} \left| \big(\bar{w}_t(i) - w(i)\big) a(i) \right| \\
&\leq \sum_{i \in [n] \setminus (A_t \cup B_t)} |\bar{w}_t(i) - w(i)| \, |a(i)| \\
&< \frac{\Delta_{(n-t+1)}}{3\text{width}(\mathcal{M})} \sum_{i \in [n] \setminus (A_t \cup B_t)} |a(i)| \qquad (2.91) \\
&= \frac{\Delta_{(n-t+1)}}{3\text{width}(\mathcal{M})} \|\boldsymbol{a}\|_1 ,
\end{aligned}$$

where Eq. (2.90) follows from the assumption that $\boldsymbol{a}$ is supported on $[n] \setminus (A_t \cup B_t)$; Eq. (2.91) follows from the definition of τ_t (Eq. (2.86)). □

2.11.2 Main Lemmas

We begin with a technical lemma which characterizes several useful properties of A_t and B_t.

Lemma 2.18. *Fix a phase $t \in [n]$. Suppose that $A_t \subseteq M_*$ and $B_t \cap M_* = \emptyset$. Let M be a set such that $A_t \subseteq M$ and $B_t \cap M = \emptyset$. Let a and b be two sets satisfying that $a \subseteq M \backslash M_*$, $b \subseteq M_* \backslash M$ and $a \cap b = \emptyset$. Then, we have*

$$A_t \subseteq (M \backslash a \cup b) \quad \textit{and} \quad B_t \cap (M \backslash a \cup b) = \emptyset \quad \textit{and} \quad (a \cup b) \cap (A_t \cup B_t) = \emptyset.$$

Proof. We prove the first part as follows

$$\begin{aligned} A_t \cap (M \backslash a \cup b) &= (A_t \cap (M \backslash a)) \cup (A_t \cap b) \\ &= A_t \cap (M \backslash a) \qquad (2.92) \\ &= (A_t \cap M) \backslash a \\ &= A_t \backslash a \qquad (2.93) \\ &= A_t, \qquad (2.94) \end{aligned}$$

where Eq. (2.92) holds since we have $A_t \cap b \subseteq A_t \cap (M_* \backslash M) \subseteq M \cap (M_* \backslash M) = \emptyset$; Eq. (2.93) follows from $A_t \subseteq M$; and Eq. (2.94) follows from $a \subseteq M \backslash M_*$ and $A_t \subseteq M_*$ which imply that $a \cap A_t = \emptyset$. Notice that Eq. (2.94) is equivalent to $A_t \subseteq (M \backslash a \cup b)$.

Then, we proceed to prove the second part in the following

$$\begin{aligned} B_t \cap (M \backslash a \cup b) &= (B_t \cap (M \backslash a)) \cup (B_t \cap b) \\ &= B_t \cap (M \backslash a) \qquad (2.95) \\ &= (B_t \cap M) \backslash a \\ &= \emptyset \backslash a = \emptyset, \qquad (2.96) \end{aligned}$$

where Eq. (2.95) follows from the fact that $B_t \cap b \subseteq B_t \cap (M_* \backslash M) \subseteq \neg M_* \cap (M_* \backslash M) = \emptyset$; and Eq. (2.96) follows from the fact that $B_t \cap M = \emptyset$.

Last, we prove the third part. By combining the assumptions that $A_t \subseteq M_*$ and $A_t \subseteq M$, we see that $A_t \subseteq M \cap M_*$. Also note that $a \subseteq M \backslash M_*$ and $b \subseteq M_* \backslash M$, we have

$$(a \cap A_t) \cup (b \cap A_t) \subseteq \big((M \backslash M_*) \cap (M \cap M_*)\big) \cup \big((M_* \backslash M) \cap (M \cap M_*)\big) = \emptyset. \tag{2.97}$$

Similarly, we have $B_t \subseteq \neg M \cap \neg M_*$. Hence, we derive

$$(a \cap B_t) \cup (b \cap B_t) \subseteq ((M \backslash M_*) \cap (\neg M \cap \neg M_*)) \cup ((M_* \backslash M) \cap (\neg M \cap \neg M_*)) = \emptyset. \tag{2.98}$$

By combining Eq. (2.97) and Eq. (2.98), we obtain

$$(a \cup b) \cap (A_t \cup B_t) = (a \cap A_t) \cup (b \cap A_t) \cup (a \cap B_t) \cup (b \cap B_t) = \emptyset.$$

□

The next lemma provides an important insight on the correctness of CSAR. Informally speaking, suppose that the algorithm does not make an error before phase t. Then, we show that, suppose arm e has a gap Δ_e larger than the "reference" gap $\Delta_{(n-t+1)}$ of phase t, then arm e must be correctly classified by M_t, i.e., $e \in M_t$ if and only if $e \in M_*$.

Lemma 2.19. *Fix any phase $t > 0$. Suppose that event τ_t occurs. Also assume that $A_t \subseteq M_*$ and $B_t \cap M_* = \emptyset$. Let $e \in [n] \backslash (A_t \cup B_t)$ be an active arm. Suppose that $\Delta_{(t-n+1)} \leq \Delta_e$. Then, we have $e \in (M_* \cap M_t) \cup (\neg M_* \cap \neg M_t)$.*

Proof. Fix an exchange class $\mathcal{B} \in \arg\min_{\mathcal{B}' \in \text{Exchange}(\mathcal{M})} \text{width}(\mathcal{B}')$. Suppose that $e \notin (M_* \cap M_t) \cup (\neg M_* \cap \neg M_t)$. This is equivalent to the following

$$e \in (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t). \tag{2.99}$$

Eq. (2.99) can be further rewritten as

$$e \in (M_* \backslash M_t) \cup (M_t \backslash M_*).$$

From this assumption, it is easy to see that $M_t \neq M_*$. Therefore we can apply Lemma 2.2. Then we know that there exists $b = (b_+, b_-) \in \mathcal{B}$ such that $e \in b_- \cup b_+$, $b_- \subseteq M_t \backslash M_*$, $b_+ \subseteq M_* \backslash M_t$, $M_t \oplus b \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e > 0$.

Using Lemma 2.18, we see that $(M_t \oplus b) \cap B_t = \emptyset$, $A_t \subseteq (M_t \oplus b)$ and $(b_+ \cup b_-) \cap (A_t \cup B_t) = \emptyset$. Now recall the definition $M_t \in \arg\max_{M \in \mathcal{M}, A_t \subseteq M, B_t \cap M = \emptyset} \bar{w}_t(M)$ and also recall that $M_t \oplus b \in \mathcal{M}$. Therefore, we obtain that

$$\bar{w}_t(M_t) \geq \bar{w}_t(M_t \oplus b). \tag{2.100}$$

On the other hand, we have

$$\begin{aligned}
\bar{w}_t(M_t \oplus b) &= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} + \boldsymbol{\chi}_b \right\rangle & (2.101)\\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle + \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle \\
&> \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle + \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{M})} \left\| \boldsymbol{\chi}_b \right\|_1 \\
& & (2.102)\\
&\geq \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle + \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_e}{3\mathrm{width}(\mathcal{M})} \left\| \boldsymbol{\chi}_b \right\|_1 \\
&\geq \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle + \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_e}{3} & (2.103)\\
&\geq \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle + \frac{2}{3}\Delta_e & (2.104)\\
&\geq \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \right\rangle = \bar{w}_t(M_t), & (2.105)
\end{aligned}$$

where Eq. (2.101) follows from Lemma 2.1; Eq. (2.102) follows from Lemma 2.17 and the fact that $(b_+ \cup b_-) \cap (A_t \cup B_t) = \emptyset$; Eq. (2.103) holds since $b \in \mathcal{B}$ which implies that $\|\boldsymbol{\chi}_b\|_1 = |b_+| + |b_-| \leq \mathrm{width}(\mathcal{B}) = \mathrm{width}(\mathcal{M})$; and Eq. (2.104) and Eq. (2.105) hold since we have shown that $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e \geq 0$.

This means that $\bar{w}_t(M_t \oplus b) > \bar{w}_t(M_t)$. This contradicts to Eq. (2.100). Therefore we have $e \in (M_* \cap M_t) \cup (\neg M_* \cap \neg M_t)$. □

The next lemma takes a step further. It shows that if $\Delta_e \geq$

$\Delta_{(n-t+1)}$ for some arm e, then the empirical gap of arm e, $\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e})$, is greater than $\frac{2}{3}\Delta_{(n-t+1)}$.

Lemma 2.20. *Fix any phase $t > 0$. Suppose that event τ_t occurs. Also assume that $A_t \subseteq M_*$ and $B_t \cap M_* = \emptyset$. Let $e \in [n] \backslash (A_t \cup B_t)$ be an active arm such that $\Delta_{(t-n+1)} \le \Delta_e$. Then, we have*

$$\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e}) > \frac{2}{3}\Delta_{(t-n+1)}.$$

Proof. By Lemma 2.19, we see that

$$e \in (M_* \cap M_t) \cup (\neg M_* \cap \neg M_t). \tag{2.106}$$

We claim that $e \in (\tilde{M}_{t,e} \backslash M_*) \cup (M_* \backslash \tilde{M}_{t,e})$ and therefore $M_* \neq \tilde{M}_{t,e}$. Recall the definition of $\tilde{M}_{t,e}$, which ensures that $e \in \tilde{M}_{t,e}$ if and only if $e \notin M_t$. By Eq. (2.106), we see that either $e \in (M_* \cap M_t)$ or $e \in (\neg M_* \cap \neg M_t)$. First let us assume that $e \in M_* \cap M_t$. Then, by definition of $\tilde{M}_{t,e}$, we see that $e \notin \tilde{M}_{t,e}$. Therefore $e \in M_* \backslash \tilde{M}_{t,e}$. On the other hand, suppose that $e \in \neg M_* \cap \neg M_t$. Then, we see that $e \in \tilde{M}_{t,e}$. This means that $e \in \tilde{M}_{t,e} \backslash M_*$.

Hence we can apply Lemma 2.2. Then we obtain that there exists $b = (b_+, b_-) \in \mathcal{B}$ such that $e \in b_+ \cup b_-$, $b_+ \subseteq M_* \backslash \tilde{M}_{t,e}$, $b_- \subseteq \tilde{M}_{t,e} \backslash M_*$, $\tilde{M}_{t,e} \oplus b \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \ge \Delta_e$.

Define $M'_{t,e} \triangleq \tilde{M}_{t,e} \oplus b$. Using Lemma 2.18, we have $A_t \subseteq M'_{t,e}$, $B_t \cap M'_{t,e} = \emptyset$ and $(b_+ \cup b_-) \cap (A_t \cup B_t) = \emptyset$. Since $M'_{t,e} \in \mathcal{M}$ and by definition $\bar{w}_t(M_t) = \max_{M \in \mathcal{M}, A_t \subseteq M, B_t \cap M = \emptyset} \bar{w}_t(M)$, we have

$$\bar{w}_t(M_t) \ge \bar{w}_t(M'_{t,e}). \tag{2.107}$$

Hence, we have

$$\begin{aligned}
\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e}) &\ge \bar{w}_t(M'_{t,e}) - \bar{w}_t(\tilde{M}_{t,e}) \\
&= \bar{w}_t(\tilde{M}_{t,e} \oplus b) - \bar{w}_t(\tilde{M}_{t,e}) \\
&= \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_{t,e}} + \boldsymbol{\chi}_b \right\rangle - \left\langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{\tilde{M}_{t,e}} \right\rangle \qquad (2.108) \\
&= \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle
\end{aligned}$$

$$> \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{B})} \|\boldsymbol{\chi}_b\|_1 \tag{2.109}$$

$$\geq \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_e}{3\mathrm{width}(\mathcal{B})} \|\boldsymbol{\chi}_b\|_1 \tag{2.110}$$

$$\geq \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle - \frac{\Delta_e}{3} \tag{2.111}$$

$$\geq \frac{2}{3}\Delta_e \geq \frac{2}{3}\Delta_{(n-t+1)}, \tag{2.112}$$

where Eq. (2.108) follows from Lemma 2.1; Eq. (2.109) follows from Lemma 2.17, the assumption on event τ_t and the fact $(b_+ \cup b_-) \cap (A_t \cup B_t) = \emptyset$; Eq. (2.110) follows from the assumption that $\Delta_e \geq \Delta_{(n-t+1)}$; Eq. (2.111) holds since $b \in \mathcal{B}$ and therefore $\|\boldsymbol{\chi}_b\|_1 \leq \mathrm{width}(\mathcal{M})$; and Eq. (2.112) follows from the fact that $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e$. □

The next lemma shows that, during phase t, if $\Delta_e \leq \Delta_{(n-t+1)}$ for some arm e, then the empirical gap of arm e is smaller than $\frac{1}{3}\Delta_{(n-t+1)}$.

Lemma 2.21. *Fix any phase $t > 0$. Suppose that event τ_t occurs. Also assume that $A_t \subseteq M_*$ and $B_t \cap M_* = \emptyset$. Suppose an active arm $e \in [n]\backslash(A_t \cup B_t)$ satisfies that $e \in (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t)$. Then, we have*

$$\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e}) \leq \frac{1}{3}\Delta_{(n-t+1)}.$$

Proof. Fix an exchange class $\mathcal{B} \in \arg\min_{\mathcal{B}' \in \mathrm{Exchange}(\mathcal{M})} \mathrm{width}(\mathcal{B}')$.

The assumption that $e \in (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t)$ can be rewritten as $e \in (M_* \backslash M_t) \cup (M_t \backslash M_*)$. This shows that $M_t \neq M_*$, hence Lemma 2.2 applies here. Therefore we know that there exists $b = (b_+, b_-) \in \mathcal{B}$ such that $e \in (b_+ \cup b_-)$, $b_+ \subseteq M_* \backslash M_t$, $b_- \subseteq M_t \backslash M_*$, $M_t \oplus b \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e > 0$.

Define $M'_{t,e} \triangleq M_t \oplus b$. We claim that

$$\bar{w}_t(\tilde{M}_{t,e}) \geq \bar{w}_t(M'_{t,e}). \tag{2.113}$$

From the definition of $\tilde{M}_{t,e}$ in Algorithm 2, we only need to show that **(a):** $e \in (M'_{t,e} \backslash M_t) \cup (M_t \backslash M'_{t,e})$ and **(b):** $A_t \subseteq M'_{t,e}$ and $B_t \cap M'_{t,e} = \emptyset$. First we prove **(a)**. Notice that $b_+ \cap b_- = \emptyset$ and $b_- \subseteq M_t$. Hence we see that $M'_{t,e} \backslash M_t = (M_t \backslash b_- \cup b_+) \backslash M_t = b_+$ and $M_t \backslash M'_{t,e} = M_t \backslash (M_t \backslash b_- \cup b_+) = b_-$. In addition, we have that $e \in (b_- \cup b_+) = (M'_{t,e} \backslash M_t) \cup (M_t \backslash M'_{t,e})$, therefore we see that **(a)** holds. Next, we notice that **(b)** follows directly from Lemma 2.18 by setting $M = M_t$. Hence we have shown that Eq. (2.113) holds.

Hence, we have

$$\begin{aligned}
\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e}) &\leq \bar{w}_t(M_t) - \bar{w}_t(M'_{t,e}) \\
&= \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} \rangle - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_{M_t} + \boldsymbol{\chi}_b \rangle \qquad (2.114) \\
&= - \langle \bar{\boldsymbol{w}}_t, \boldsymbol{\chi}_b \rangle \\
&\leq - \langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle + \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{M})} \|\boldsymbol{\chi}_b\|_1 \qquad (2.115) \\
&\leq \frac{\Delta_{(n-t+1)}}{3\mathrm{width}(\mathcal{M})} \|\boldsymbol{\chi}_b\|_1 \leq \frac{\Delta_{(n-t+1)}}{3}, \qquad (2.116)
\end{aligned}$$

where Eq. (2.114) follows from Lemma 2.1; Eq. (2.115) follows from Lemma 2.17, the assumption on τ_t and $(b_+ \cup b_-) \cap (A_t \cup B_t) = \emptyset$ (by Lemma 2.18); and Eq. (2.116) follows from the fact $\|\boldsymbol{\chi}_b\|_1 \leq \mathrm{width}(\mathcal{M})$ (since $b \in \mathcal{B}$) and that $\langle \boldsymbol{w}, \boldsymbol{\chi}_b \rangle \geq \Delta_e \geq 0$. □

2.11.3 Proof of Theorem 2.3

Using these technical lemmas, we are now ready to prove Theorem 2.3. For reader's convenience, we first restate Theorem 2.3 as follows.

Theorem 2.3. *Given any $T > n$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\boldsymbol{w} \in \mathbb{R}^n$. Assume that the reward distribution φ_e for each arm $e \in [n]$ has mean $w(e)$ with an R-sub-Gaussian tail. Let $\Delta_{(1)}, \ldots, \Delta_{(n)}$ be a permutation of $\Delta_1, \ldots, \Delta_n$*

(defined in Eq. (2.1)*) such that* $\Delta_{(1)} \leq \ldots\ldots \Delta_{(n)}$. *Define* $\mathbf{H}_2 \triangleq \max_{i\in[n]} i\Delta_{(i)}^{-2}$. *Then, the* CSAR *algorithm uses at most* T *samples and outputs a solution* $\mathsf{Out} \in \mathcal{M} \cup \{\perp\}$ *such that*

$$\Pr[\mathsf{Out} \neq M_*] \leq n^2 \exp\left(-\frac{(T-n)}{18R^2\tilde{\log}(n)\mathit{width}(\mathcal{M})^2\mathbf{H}_2}\right), \quad (2.8)$$

where $\tilde{\log}(n) \triangleq \sum_{i=1}^n i^{-1}$, $M_* = \arg\max_{M\in\mathcal{M}} w(M)$ *and* $\mathit{width}(\mathcal{M})$ *is defined in Eq.* (2.4)*.*

Proof. First, we show that the algorithm takes at most T samples. It is easy to see that exactly one arm is pulled for $\tilde{T}_1$ times, one arm is pulled for $\tilde{T}_2$ times, ..., and one arm is pulled for $\tilde{T}_n$ times. Therefore, the total number of samples used by the algorithm is bounded by

$$\sum_{t=1}^n \tilde{T}_t \leq \sum_{t=1}^n \left(\frac{T-n}{\tilde{\log}(n)(n-t+1)} + 1\right)$$
$$= \frac{T-n}{\tilde{\log}(n)}\tilde{\log}(n) + n = T.$$

By Lemma 2.16, we know that the event $\tau \triangleq \bigcap_{t=1}^T \tau_t$ occurs with probability at least $1 - n^2 \exp\left(-\frac{(T-n)}{18R^2\tilde{\log}(n)\mathrm{width}(\mathcal{M})^2\mathbf{H}_2}\right)$. Therefore, we only need to prove that, under event τ, the algorithm outputs M_*. We will assume that event τ occurs in the rest of the proof.

We prove by induction. Fix a phase $t \in [t]$. Suppose that the algorithm does not make any error before phase t, i.e., $A_t \subseteq M_*$ and $B_t \cap M_* = \emptyset$. We show that the algorithm does not err at phase t.

At the beginning of phase t, there are exactly $t-1$ inactive arms $|A_t \cup B_t| = t-1$. Therefore there must exists an active arm $e_t \in [n]\backslash(A_t \cup B_t)$ such that $\Delta_{e_t} \geq \Delta_{(n-t+1)}$. Hence, by Lemma 2.20, we have

$$\bar{w}_t(M_t) - \bar{w}_t(M_{t,e_t}) \geq \frac{2}{3}\Delta_{(n-t+1)}. \quad (2.117)$$

Notice that the algorithm makes an error in phase t if and only if it accepts an arm $p_t \notin M_*$ or rejects an arm $p_t \in M_*$. On the other hand, by design, arm p_t is accepted when $p_t \in M_t$ and is rejected when $p_t \notin M_t$. Therefore, we see that the algorithm makes an error in phase t if and only if $p_t \in (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t)$.

Suppose that $p_t \in (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t)$. Now appeal to Lemma 2.21, we see that

$$\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,p_t}) \leq \frac{1}{3}\Delta_{(n-t+1)}. \tag{2.118}$$

By combining Eq. (2.117) and Eq. (2.118), we see that

$$\bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,p_t}) \leq \frac{1}{3}\Delta_{(n-t+1)} < \frac{2}{3}\Delta_{(n-t+1)} \leq \bar{w}_t(M_t) - \bar{w}_t(M_{t,e_t}). \tag{2.119}$$

However Eq. (2.119) is contradictory to the definition of $p_t \triangleq \arg\max_{e \in [n] \setminus (A_t \cup B_t)} \bar{w}_t(M_t) - \bar{w}_t(\tilde{M}_{t,e})$. Therefore we have proven that $p_t \notin (M_* \cap \neg M_t) \cup (\neg M_* \cap M_t)$. This means that the algorithm does not err at phase t, or equivalently $A_{t+1} \subseteq M_*$ and $B_{t+1} \cap M_* = \emptyset$. By induction, we have proven that the algorithm does not err at any phase $t \in [n]$.

Hence we have $A_{n+1} \subseteq M_*$ and $B_{n+1} \subseteq \neg M_*$ in the final phase. Notice that $|A_{n+1}| + |B_{n+1}| = n$ and $A_{n+1} \cap B_{n+1} = \emptyset$. This means that $A_{n+1} = M_*$ and $B_{n+1} = \neg M_*$. Therefore the algorithm outputs $\mathsf{Out} = A_{n+1} = M_*$ after phase n. □

2.12 Analysis of the Uniform Allocation Algorithm

In this section, we analyze the performance of a simple benchmark strategy UNI which plays each arm for a equal number of times and then calls a maximization oracle using the empirical means of arms as input. The pseudo-code of the UNI algorithm is listed in Algorithm 3.

```
Algorithm 3 UNI: Uniform Allocation
Require: Budget: $T > 0$; Maximization oracle: Oracle : $\mathbb{R}^n \to \mathcal{M}$.
Pull each arm $e \in [n]$ for $\lfloor T/n \rfloor$ times.
Compute the empirical means $\bar{\boldsymbol{w}} \in \mathbb{R}^n$ of each arm.
Out $\leftarrow$ Oracle($\bar{\boldsymbol{w}}$)
return: Out
```

The next theorem upper bounds the probability of error of UNI.

Theorem 2.7. *Given any $T > n$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\boldsymbol{w} \in \mathbb{R}^n$. Assume that the reward distribution φ_e for each arm $e \in [n]$ has mean $w(e)$ with an R-sub-Gaussian tail. Also assume without loss of generality that T is a multiple of n. Define $\Delta_{(1)} = \min_{i\in[n]} \Delta_i$ and $\mathbf{H}_3 = n\Delta_{(1)}^{-2}$. Then, the output* Out *of the* UNI *algorithm satisfies*

$$\Pr[\mathsf{Out} \neq M_*] \leq 2n \exp\left(-\frac{T}{18R^2 width(\mathcal{M})^2 \mathbf{H}_3}\right), \qquad (2.120)$$

where $M_* = \arg\max_{M\in\mathcal{M}} w(M)$.

From Theorem 2.7, we see that the UNI algorithm could be significantly worse than CLUCB and CSAR, since it is clear that $\mathbf{H}_3 \geq \mathbf{H} \geq \mathbf{H}_2$ and potentially one has $\mathbf{H}_3 \gg \mathbf{H} \geq \mathbf{H}_2$ for a large number of arms with heterogeneous gaps.

Now we prove Theorem 2.7. The proof is straightforward using tools of exchange classes.

Proof. Define $\Delta_{(1)} = \min_{i\in[n]} \Delta_i$. Define random event ξ as follows

$$\xi = \left\{\forall i \in [n], \quad |\bar{w}(i) - w(i)| < \frac{\Delta_{(1)}}{3\text{width}(\mathcal{M})}\right\}.$$

Notice that each arm is sampled for $\lfloor \frac{T}{n} \rfloor$ times. Therefore, using Hoeffding's inequality (Lemma 2.6) and union bound, we can bound $\Pr[\xi]$ as follows. Fix any $i \in [n]$, by Hoeffding's inequality, we have

$$\Pr\left[|\bar{w}(i) - w(i)| \geq \frac{\Delta_{(1)}}{3\mathrm{width}(\mathcal{M})}\right] \leq 2\exp\left(-\frac{T\Delta_{(1)}^2}{18R^2 n \mathrm{width}(\mathcal{M})^2}\right).$$

Then, using a union bound, we obtain

$$\Pr\left[\xi\right] \geq 1 - 2n\exp\left(-\frac{T\Delta_{(1)}^2}{18nR^2\mathrm{width}(\mathcal{M})^2}\right).$$

In addition, using an argument very similar to Lemma 2.17, one can show that, on event ξ, for any vector $\boldsymbol{a} \in \mathbb{R}^n$, it holds that

$$\left|\langle \bar{\boldsymbol{w}}, \boldsymbol{a}\rangle - \langle \boldsymbol{w}, \boldsymbol{a}\rangle\right| < \frac{\Delta_{(1)}}{3\mathrm{width}(\mathcal{M})}\|\boldsymbol{a}\|_1. \tag{2.121}$$

Now we claim that, on the event ξ, we have $\mathsf{Out} = M_*$. Note that theorem follows immediately from the claim. Next, we prove this claim.

Suppose that, on the contrary, $\mathsf{Out} \neq M_*$. In this case, let us write $M = \mathsf{Out}$. We also fix $\mathcal{B} \in \arg\min_{\mathcal{B}' \in \mathrm{Exchange}(\mathcal{M})} \mathrm{width}(\mathcal{B})$. Notice that by definition $\mathrm{width}(\mathcal{B}) = \mathrm{width}(\mathcal{M})$.

Since $M \neq M_*$, we see that there exists $e \in (M \backslash M_*) \cup (M_* \backslash M)$. Now, by Lemma 2.2, we obtain that there exists $b = (b_+, b_-) \in \mathcal{B}$ such that $e \in b_+ \cup b_-$, $b_- \subseteq M \backslash M_*$, $b_+ \subseteq M_* \backslash M$, $M \oplus b \in \mathcal{M}$ and $\langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle \geq \Delta_e$. Also notice that $\Delta_e \geq \Delta_{(1)}$. Therefore $\langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle \geq \Delta_{(1)}$.

Consider $M' \triangleq M \oplus b$. We have

$$\begin{aligned}
\bar{w}(M') - \bar{w}(M) &= \langle \bar{\boldsymbol{w}}, \boldsymbol{\chi}_{M'}\rangle - \langle \bar{\boldsymbol{w}}, \boldsymbol{\chi}_M\rangle \\
&= \langle \bar{\boldsymbol{w}}, \boldsymbol{\chi}_b\rangle && (2.122) \\
&> \langle \boldsymbol{w}, \boldsymbol{\chi}_b\rangle - \frac{\Delta_{(1)}}{3\mathrm{width}(\mathcal{M})}\|\boldsymbol{\chi}_b\|_1 && (2.123) \\
&\geq \Delta_{(1)} - \frac{\Delta_{(1)}}{3} && (2.124)
\end{aligned}$$

$$= \frac{2}{3}\Delta_{(1)} > 0, \tag{2.125}$$

where Eq. (2.122) follows from Lemma 2.1; Eq. (2.123) follows from Eq. (2.121); and Eq. (2.124) follows from the fact that $b \in \mathcal{B}$ and hence $\|\boldsymbol{\chi}_b\|_1 = |b_+| + |b_-| \leq \mathrm{width}(\mathcal{B}) = \mathrm{width}(\mathcal{M})$.

Hence, we have shown that $\bar{w}(M') > \bar{w}(M)$. However this contradicts to the fact that $\bar{w}(M) = \max_{M_1 \in \mathcal{M}} \bar{w}(M_1)$ (by the definition of maximization oracle). Hence, by contradiction, we have proven that $\mathsf{Out} = M_*$. □

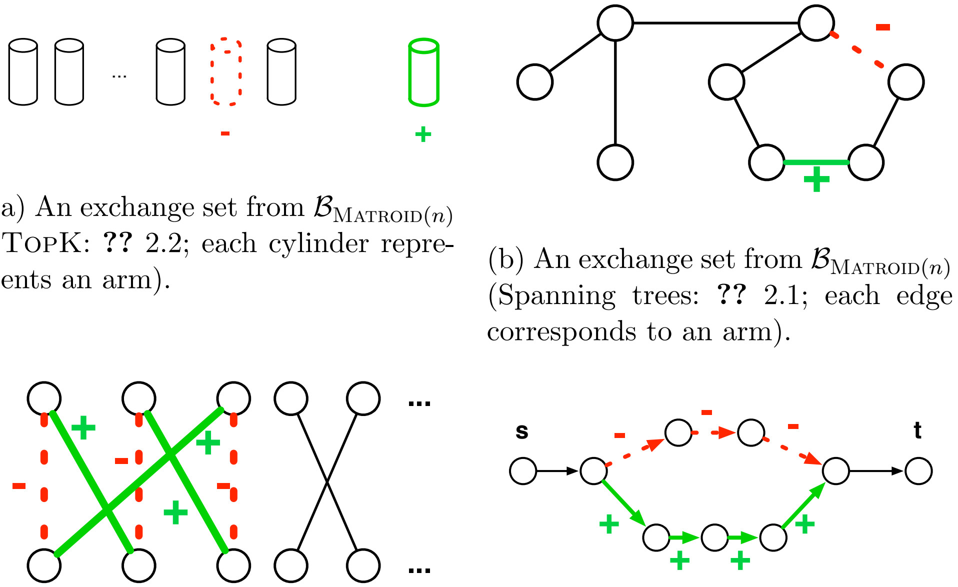
2.13 Exchange Classes for Example Decision Classes

In this section, we give formal constructions of the decision classes discussed in Example 1, 2 and 3. Further, we bound the width of exchange classes for different examples. These bounds are proven using concrete constructions of exchange classes (Fact 2.1 through 2.5). The constructed exchange classes embody natural combinatorial structures. We illustrate the constructed exchange classes in Figure 2.3.

Notation. We need one extra notation. Let $\sigma : E \to [n]$ be a bijection from some set E with n elements to $[n]$. Let $A \subseteq E$ be an arbitrary set, we define $\sigma(A) \triangleq \{\sigma(a) \mid a \in A\}$. Conversely, for all $M \subseteq [n]$, we define $\sigma^{-1}(M) \triangleq \{\sigma^{-1}(e) \mid e \in M\}$.

Fact 2.1 (Matroid). *Let $T = (E, \mathcal{I})$ be an arbitrary matroid, where E is the ground set of n elements and $\mathcal{I}$ is the family of subsets of E called in the independent sets which satisfy the axioms of matroids* [3]. *Let $\sigma\colon E \to [n]$ be a bijection from E to*

[3] The three axioms of matroid are (1) $\emptyset \in \mathcal{I}$ and $\mathcal{I} \neq \{\emptyset\}$; (2) Every subsets of an independent set are independent (heredity property); (3) For all $A, B \in \mathcal{I}$ such that $|B| = |A| + 1$ there exists an element $e \in B \backslash A$ such that $A \cup \{e\} \in \mathcal{I}$ (augmentation property). We refer interested readers to [112] for a general introduction to the matroid theory.



(a) An exchange set from $\mathcal{B}_{\text{MATROID}(n)}$ (TopK: **??** 2.2; each cylinder represents an arm).

(b) An exchange set from $\mathcal{B}_{\text{MATROID}(n)}$ (Spanning trees: **??** 2.1; each edge corresponds to an arm).

(c) An exchange set from $\mathcal{B}_{\text{MATCH}(G)}$ (Matchings: **??** 2.4; each edge corresponds to an arm)

(d) An exchange set from $\mathcal{B}_{\text{PATH}(G)}$ (Paths: **??** 2.5; each edge corresponds to an arm).

Figure 2.3: Examples of exchange sets belonging to the exchange classes $\mathcal{B}_{\text{MATROID}(n)}$ (one for TopK and one for spanning tree), $\mathcal{B}_{\text{MATCH}(G)}$ and $\mathcal{B}_{\text{PATH}(G)}$: green-solid elements constitute the set b_+, red-dotted elements constitute the set b_- and an example exchange set is given by $b = (b_+, b_-)$. In Figure 2.3a, we use TopK as a specific instance of matroid decision class. In Figure 2.3b, we use spanning tree as a specific instance of matroid decision class.

$[n]$. *Let* $\mathcal{M}_{\text{MATROID}(T)}$ *correspond to the collection of all bases of matroid* T *and formally we define*

$$\mathcal{M}_{\text{MATROID}(T)} = \left\{ M \subseteq [n] \mid \sigma^{-1}(M) \textit{ is a basis of } T \right\}. \tag{2.126}$$

Define the exchange class

$$\mathcal{B}_{\text{MATROID}(n)} = \left\{ (\{i\}, \{j\}) \mid \forall i \in [n], j \in [n] \right\}. \tag{2.127}$$

Then we have $\mathcal{B}_{\text{MATROID}(n)} \in \text{Exchange}(\mathcal{M}_{\text{MATROID}(T)})$. *In addition, we have width*$(\mathcal{B}_{\text{MATROID}(n)}) = 2$*, which implies that width*$(\mathcal{M}_{\text{MATROID}(T)}) \leq 2$.

To prove Fact 2.1, we first recall a well-known result from matroid theory which is referred as the strong basis exchange property.

Lemma 2.22 (Strong basis exchange [112]). *Let $\mathcal{A}$ be the set of all bases of a matroid $T = (E, \mathcal{I})$. Let $A_1, A_2 \in \mathcal{A}$ be two bases. Then for all $x \in A_1 \backslash A_2$, there exists $y \in A_2 \backslash A_1$ such that $A_1 \backslash \{x\} \cup \{y\} \in \mathcal{A}$ and $A_2 \backslash \{y\} \cup \{x\} \in \mathcal{A}$.*

Using Lemma 2.22, we are ready to prove Fact 2.1.

Proof of Fact 2.1. Fix a matroid $T = (E, \mathcal{I})$ where $|E| = n$ and fix the bijection $\sigma \colon E \to [n]$. Let $\mathcal{M}_{\textsc{Matroid}(T)}$ be defined as in Eq. (2.126) and let $\mathcal{B}_{\textsc{Matroid}(n)}$ be defined as in Eq. (2.127). Let $\mathcal{A}$ denote the set of all bases of T. By definition, we have $\mathcal{M}_{\textsc{Matroid}(T)} = \{\sigma(A) \mid A \in \mathcal{A}\}$.

Now we show that $\mathcal{B}_{\textsc{Matroid}(n)}$ is an exchange class for $\mathcal{M}_{\textsc{Matroid}(T)}$. Let M, M' be two different elements of $\mathcal{M}_{\textsc{Matroid}(T)}$. By definition, we see that $\sigma^{-1}(M)$ and $\sigma^{-1}(M')$ are two bases of T. Consider any $e \in M \backslash M'$. Let $x = \sigma^{-1}(e)$. We see that $x \in \sigma^{-1}(M) \backslash \sigma^{-1}(M')$.

By Fact 2.22, we see that there exists $y \in \sigma^{-1}(M') \backslash \sigma^{-1}(M)$ such that

$$\sigma^{-1}(M) \backslash \{x\} \cup \{y\} \in \mathcal{A} \quad \text{and} \quad \sigma^{-1}(M') \backslash \{y\} \cup \{x\} \in \mathcal{A}. \tag{2.128}$$

Now we define exchange set $b = (b_+, b_-)$ where $b_+ = \{\sigma(y)\}$ and $b_- = \{\sigma(x)\}$. By Eq. (2.128) and the fact that σ is a bijection, we see that $M \oplus b \in \mathcal{M}_{\textsc{Matroid}(T)}$ and $M' \ominus b \in \mathcal{M}_{\textsc{Matroid}(T)}$. We also have $b \in \mathcal{B}_{\textsc{Matroid}(n)}$. Due to M, M' and e are chosen arbitrarily, we have verified that $\mathcal{B}_{\textsc{Matroid}(n)}$ is an exchange class for $\mathcal{M}_{\textsc{Matroid}(T)}$.

To conclude, we observe that $\text{width}(\mathcal{B}_{\textsc{Matroid}(n)}) = 2$. □

Now we show that TopK and MB are special cases of the family of decision classes of derived from bases of matroids. This enable us to apply Fact 2.1 to construct exchange classes and bound the widths of these decision classes. We also note that one may use a more direct way to construct the exchange classes for these two problems without appealing to matroids.

Fact 2.2 (TopK). *For all $K \in [n]$, let $\mathcal{M}_{\textsc{TopK}(K)} = \{M \subseteq [n] \mid |M| = K\}$ be the collection of all subsets of size K. Then we have $\mathcal{B}_{\textsc{Matroid}(n)} \in \mathrm{Exchange}(\mathcal{M}_{\textsc{TopK}(K)})$ and width$(\mathcal{M}_{\textsc{TopK}(K)}) \leq 2$.*

Proof. Let $U_n^K = ([n], \mathcal{I}_K)$ where $\mathcal{I}_K$ is given by

$$\mathcal{I}_K = \big\{M \subseteq [n] \mid |M| \leq K\big\}.$$

Recall that U_n^K is a matroid (in particular, a uniform matroid of rank K) [112]. We know that a subset M of $[n]$ is basis of U_n^K if and only if $|M| = K$. Therefore, we have $\mathcal{M}_{\textsc{TopK}(K)} = \mathcal{M}_{\textsc{Matroid}(U_n^K)}$. Then we can conclude immediately by using **??** 2.1. □

Fact 2.3 (MB). *For any partition $\mathcal{A} = \{A_1, \ldots, A_m\}$ of $[n]$, we define*

$$\mathcal{M}_{\mathrm{MB}(\mathcal{A})} = \Big\{M \subseteq [n] \mid \forall i \in [m] \quad |M \cap A_i| = 1\Big\}.$$

Then we have $\mathcal{B}_{\textsc{Matroid}(n)} \in \mathrm{Exchange}(\mathcal{M}_{\textsc{TopK}(K)})$ and width$(\mathcal{M}_{\mathrm{MB}(\mathcal{A})}) \leq 2$.

Proof. Let $P_{\mathcal{A}} = ([n], \mathcal{I}_{\mathcal{A}})$ where $\mathcal{I}_{\mathcal{A}}$ is given by

$$\mathcal{I}_{\mathcal{A}} = \big\{M \subseteq [n] \mid \forall i \in [m] \quad |M \cap A_i| \leq 1\big\}.$$

It can be shown that $P_{\mathcal{A}}$ is a matroid (known as partition matroid [112]) and each basis M of $P_{\mathcal{A}}$ satisfies $|M \cap A_i| = 1$ for all $i \in [m]$. Therefore we have $\mathcal{M}_{\mathrm{MB}(\mathcal{A})} = \mathcal{M}_{\textsc{Matroid}(P_{\mathcal{A}})}$. Then the conclusion follows immediately from Fact 2.1. □

Fact 2.4 (Matching)**.** *Let $G(V, E)$ be a bipartite graph with n edges. Let $\sigma\colon E \to [n]$ be a bijection. Let $\mathcal{A}$ be the set of all valid matchings in G. We define $\mathcal{M}_{\textsc{Match}(G)}$ as follows*

$$\mathcal{M}_{\textsc{Match}(G)} = \big\{\sigma(A) \mid A \in \mathcal{A}\big\}.$$

Define the exchange class

$$\mathcal{B}_{\textsc{Match}(G)} = \Big\{(\sigma(c_+), \sigma(c_-)) \mid \exists c \in \mathcal{C} \cup \mathcal{P},\ \textit{the edges of } c \textit{ alternate between } c_+, c_-\Big\},$$

where $\mathcal{C}$ is the set of all cycles in G and $\mathcal{P}$ is the set of all paths in G. Then we have $\mathcal{B}_{\textsc{Match}(G)} \in \mathrm{Exchange}(\mathcal{M}_{\textsc{Match}(G)})$. In addition, we have width$(\mathcal{B}_{\textsc{Match}(G)}) \le |V|$, which implies that width$(\mathcal{M}_{\textsc{Matroid}(T)}) \le |V|$.

To prove Fact 2.4, we recall a classical result on graph matching which characterizes the properties of augmenting cycles and augmenting paths [18].

Lemma 2.23. *Let $G(V, E)$ be a bipartite graph. Let M and M' be two different matchings of G. Then the induced graph G' from the symmetric difference $(M \backslash M') \cup (M' \backslash M)$ consists of connected components that are one of the following*

- *An even cycle whose edges alternate between M and M'.*
- *A simple path whose edges alternate between M and M'.*

Proof of Fact 2.4. Fix a bipartite graph $G(V, E)$ and a bijection $\sigma\colon E \to [n]$. Let $M, M' \in \mathcal{M}_{\textsc{Match}(G)}$ be two different elements of $\mathcal{M}_{\textsc{Match}(G)}$ and consider an arbitrary $e \in M \backslash M'$. On a high level perspective, we construct an exchange class which contains all augmenting cycles and paths of G. We know that the symmetric difference between M and M' can be decomposed into a collection of disjoint augmenting cycles and paths. And e must be on one of the augmenting cycle or path. Then, since "applying" this augmenting cycle/path on M will yield another valid matching which does not contains e. We see that this meets the requirements of an exchange class. In the rest of the proof, we carry out the technical details of this argument.

Define $A = \sigma^{-1}(M)$ and $A' = \sigma^{-1}(M')$. Let $a = \sigma^{-1}(e)$. Then A, A' are two matchings of G. Let G' be the induced graph from the symmetric difference $(A \backslash A') \cup (A' \backslash A)$. Let C be the connected component of G' which contains the edge a. Therefore, by Lemma 2.23, we see that C is either an even cycle or a simple path with edges alternating between A and A'. Let C_+ contains the edges of C that belongs to $A' \backslash A$. Similarly, let C_- contains the edges of C that belongs to $A \backslash A'$. Define $b_+ = \sigma(C_+)$ and $b_- = \sigma(C_-)$. Let $b = (b_+, b_-)$ be an exchange set.

Since b corresponds to either an augmenting path or an augmenting cycle, we see that $b \in \mathcal{B}_{\text{MATCH}(G)}$. Since $a \in C_-$, we obtain that $e \in b_-$. In addition, note that $C_+ \subseteq A' \backslash A$ and $C_- \subseteq A \backslash A'$. Therefore we have $b_+ \subseteq M' \backslash M$ and $b_- \subseteq M \backslash M'$.

Since C is an A-augmenting path/cycle, therefore it immediately holds that $A \backslash C_- \cup C_+$ is a valid matching. Therefore, we have $M \backslash b_- \cup b_+ \in \mathcal{M}_{\text{MATCH}(G)}$. Similarly, one can show that $M' \backslash b_+ \cup b_- \in \mathcal{M}_{\text{MATCH}(G)}$. Hence we have shown that $\mathcal{B}_{\text{MATCH}(G)}$ is an exchange class for $\mathcal{M}_{\text{MATCH}(G)}$. □

Fact 2.5 (Path). *Let $G(V, E)$ be a directed acyclic graph with n edges. Let $s, t \in V$ be two different vertices. Let $\sigma \colon E \to [n]$ be a bijection. Let $\mathcal{A}(s, t)$ be the set of all valid paths from s to t in G. We define $\mathcal{M}_{\text{PATH}(G,s,t)}$ as follows*

$$\mathcal{M}_{\text{PATH}(G,s,t)} = \big\{\sigma(A) \mid A \in \mathcal{A}(s,t)\big\}.$$

Define exchange class

$$\mathcal{B}_{\text{PATH}(G)} = \{(\sigma^{-1}(p), \sigma^{-1}(q)) \mid p, q \textit{ are the arcs of two disjoint paths of } G \textit{ with sam}$$

Then, we have $\mathcal{B}_{\text{PATH}(G)} \in$ Exchange$(\mathcal{M}_{\text{PATH}(G,s,t)})$. In addition, we have width$(\mathcal{B}_{\text{PATH}(G)}) \leq |V|$ and therefore width$(\mathcal{M}_{\text{PATH}(G,s,t)}) \leq |V|$.

Proof. Fix a directed acyclic graph $G(V, E)$ and a bijection $\sigma\colon E \to [n]$. Fix two vertices $s, t \in V$.

We prove that $\mathcal{B}_{\textsc{Path}(G)}$ is an exchange class for $\mathcal{M}_{\textsc{Path}(G,s,t)}$. Let $M, M' \in \mathcal{M}_{\textsc{Path}(G,s,t)}$ be two different sets. Then $\sigma^{-1}(M), \sigma^{-1}(M')$ are two sets of arcs corresponding to two different paths from s to t. Let $P = (v_1, \ldots, v_{n_1}), P' = (v'_1, \ldots, v'_{n_2})$ denote the two paths, respectively. Notice that $s = v_1 = v'_1$ and $t = v_{n_1} = v'_{n_2}$. We also denote $E(P) = \sigma^{-1}(M)$ and $E(P') = \sigma^{-1}(M')$.

Fix some $e \in M \backslash M'$ and define $a = \sigma^{-1}(e)$. Suppose that a is an arc from u to v. Since a is on path P, there exists i such that $v_i = u$ and $v_{i+1} = v$. Now we define $j_1 = \arg\max_{j \leq i, v_j \in P'} j$ and $j_2 = \arg\min_{j \geq i+1, v_j \in P'} j$. Notice that j_1 and j_2 are well-defined since P and P' intersects on at least two vertices (s and t). Let $v'_{k_1} = v_{j_1}$ and $v'_{k_2} = v_{j_2}$ be the corresponding indices in P'. Then, we see that $Q_1 = (v_{j_1}, v_{j_1+1}, \ldots, v_{j_2})$ and $Q_2 = (v'_{k_1}, v'_{k_1+1}, \ldots, v'_{k_2})$ are two different paths from v_{j_1} to v_{j_2}. Denote the sets of arcs of Q_1 and Q_2 as $E(Q_1)$ and $E(Q_2)$.

Let $b = (b_+, b_-)$, where $b_+ = \sigma(E(Q_2)), b_- = \sigma(E(Q_1))$. We see that $b \in \mathcal{B}_{\textsc{Path}(G)}$. It is clear that $a \in E(Q_1)$, $E(Q_1) \subseteq E(P) \backslash E(P')$ and $E(Q_2) \subseteq E(P') \backslash E(P)$. Therefore $e \in b_-$, $b_- \subseteq M \backslash M'$ and $b_+ \subseteq M' \backslash M$.

Now it is easy to check that $E(P_1) \backslash E(Q_1) \cup E(Q_2)$ equals the set of arcs of path $(v_1, \ldots, v_{j_1}, v'_{k_1+1}, \ldots, v'_{k_2-1}, v_{j_2}, \ldots, v_{n_1})$ (recall that $v_{j_1} = v'_{k_1}$ and $v_{j_2} = v'_{k_2}$). This means that $E(P_1) \backslash E(Q_1) \cup E(Q_2) \in \mathcal{A}(s, t)$ and therefore $M \backslash b_- \cup b_+ \in \mathcal{M}_{\textsc{Path}(G,s,t)}$. Using a similar argument, one can show that $M' \backslash b_+ \cup b_- \in \mathcal{M}_{\textsc{Path}(G,s,t)}$ and hence we have verified that $\mathcal{B}_{\textsc{Path}(G)} \in \text{Exchange}(\mathcal{M}_{\textsc{Path}(G,s,t)})$. □

2.14 Equivalence Between Constrained Oracles and Maximization Oracles

In this section, we present a general method to implement constrained oracles using maximization oracles. The idea of the reduction is simple: one can impose the negative constrains B by setting the corresponding weights to be sufficiently small; and one can impose the positive constrains A by setting the corresponding weights to be sufficiently large. The reduction method is shown in Algorithm 4. The correctness of the reduction is proven in Lemma 2.24. Furthermore, it is trivial to reduce from maximization oracles to constrained oracles. Therefore, Lemma 2.24 shows that maximization oracles are equivalent to constrained oracles up to a transformation on the weight vector.

Lemma 2.24. *Given $\mathcal{M} \subseteq 2^{[n]}$, $\boldsymbol{w} \in \mathbb{R}^n$, $A \subseteq [n]$ and $B \subseteq [n]$, suppose that $A \cap B = \emptyset$. Then the output* Out *of Algorithm 4 satisfies* $\mathsf{Out} \in \arg\max_{M \in \mathcal{M}, A \subseteq M, B \cap M = \emptyset} w(M)$ *where we use the convention that the* $\arg\max$ *of an empty set is $\perp$. Therefore Algorithm 4 is a valid constrained oracle.*

Proof. Let $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ be defined as in Algoritm 4. Let $M = \text{Oracle}(\boldsymbol{w}_2)$. Let $\mathcal{M}_{A,B} = \{M \in \mathcal{M} \mid A \subseteq M, B \cap M = \emptyset\}$ be the subset of $\mathcal{M}$ which satisfies the constraints. If $\mathcal{M}_{A,B} = \emptyset$, then it is clear M cannot satisfy both of the constraints $A \subseteq M$ and $B \cap M = \emptyset$. Therefore Algorithm 4 returns $\perp$ in this case.

In the rest of the proof, we assume that $\mathcal{M}_{A,B} \neq \emptyset$. Since $\mathcal{M}_{A,B}$ is non-empty, we can fix an arbitrary $M_0 \in \mathcal{M}_{A,B}$, which will be used later in the proof. We will also frequently use the fact that, for all $\boldsymbol{v} \in \mathbb{R}^n$ and all $S \subseteq [n]$, we have

$$-\left\|\boldsymbol{v}\right\|_1 \leq v(S) \leq \left\|\boldsymbol{v}\right\|_1. \tag{2.129}$$

First we claim that $B \cap M = \emptyset$. Suppose that $B \cap M \neq \emptyset$.

Algorithm 4 COracle($\boldsymbol{w}, A, B$)

```
Require: $\boldsymbol{w} \in \mathbb{R}^n$, $A \subseteq [n]$, $B \subseteq [n]$; Maximization oracle Oracle : $\mathbb{R}^n \to \mathcal{M}$
$L_1 \leftarrow \|\boldsymbol{w}\|_1$
for $i = 1, \dots, n$ do
    if $i \in A$ then
        $w_1(i) \leftarrow 3L_1$
    else
        $w_1(i) \leftarrow w(i)$
    end if
end for
$L_2 \leftarrow \|\boldsymbol{w}_1\|_1$
for $i = 1, \dots, n$ do
    if $i \in B$ then
        $w_2(i) \leftarrow -3L_2$
    else
        $w_2(i) \leftarrow w_1(i)$
    end if
end for
$M \leftarrow$ Oracle($\boldsymbol{w}_2$)
if $B \cap M = \emptyset$ and $A \subseteq M$ then
    Out $= M$
else
    Out $= \perp$
end if
return: Out
```

Then there exists $i \in B \cap M$ and we fix such an i. Then we have

$$\begin{aligned} w_2(M) &= w_2(M \backslash \{i\}) + w_2(i) \\ &\leq w_2(M \backslash B) + w_2(i) && (2.130) \\ &= w_1(M \backslash B) + w_2(i) && (2.131) \\ &\leq L_2 - 3L_2 = -2L_2, && (2.132) \end{aligned}$$

where Eq. (2.130) follows from the fact that $w_2(j) = -L_2 \leq 0$ for all $j \in B \backslash \{i\}$; Eq. (2.131) holds since $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ coincide on all entries of $M \backslash B$; and Eq. (2.132) follows from the definition $L_2 = \|\boldsymbol{w}_1\|_1$ and Eq. (2.129).

On the other hand, observing that $B \cap M_0 = \emptyset$, we can bound

$w_2(M_0)$ as follows

$$w_2(M_0) = w_1(M_0) \geq -L_2.$$

Therefore we see that $w_2(M_0) > w_2(M)$. However, this contradicts to the definition of M since $M \in \arg\max_{M' \in \mathcal{M}} w_2(M')$. Therefore our claim $B \cap M = \emptyset$ is true. By this claim and since $\boldsymbol{w}_2$ and $\boldsymbol{w}_1$ coincide on entries of $[n] \backslash B$, we have

$$w_2(M) = w_1(M). \tag{2.133}$$

Next we claim that $A \subseteq M$. Suppose that $A \not\subseteq M$. Then we have

$$\begin{aligned} w_2(M) = w_1(M) &= w_1(M \cap A) + w_1(M \backslash A) \\ &= 3|M \cap A| L_1 + w(M \backslash A) \qquad (2.134) \\ &\leq (3|A| - 3) L_1 + L_1 \qquad (2.135) \\ &= (3|A| - 2) L_1, \qquad (2.136) \end{aligned}$$

where Eq. (2.134) follows from the definition of $\boldsymbol{w}_1$; and Eq. (2.135) follows from the assumption that $A \not\subseteq M$ and therefore $|M \cap A| \leq |A| - 1$.

On the other hand, using the fact that $A \subseteq M_0$ (since $M_0 \in \mathcal{M}_{A,B}$), we have

$$\begin{aligned} w_2(M_0) = w_1(M_0) &= w_1(A) + w_1(M_0 \backslash A) \qquad (2.137) \\ &= 3|A| L_1 + w(M_0 \backslash A) \qquad (2.138) \\ &\geq 3|A| L_1 - L_1 \qquad (2.139) \\ &= (3|A| - 1) L_1, \qquad (2.140) \end{aligned}$$

where Eq. (2.137) follows from the fact that $M_0 \cap B = \emptyset$ and $A \subseteq M_0$; Eq. (2.138) follows from the definition of $\boldsymbol{w}_1$, which ensures that $\boldsymbol{w}_1$ and $\boldsymbol{w}$ coincide on $M_0 \backslash A$; and Eq. (2.139) follows from Eq. (2.129).

Therefore, by combining Eq. (2.136) and Eq. (2.140), we see that $w_2(M_0) > w_2(M)$. Again this contradicts to the definition of M, which proves the claim that $A \subseteq M$.

Now we see that $A \subseteq M$ and $B \cap M = \emptyset$, which means that $M \in \mathcal{M}_{A,B}$. Therefore, we remain to verify that $w(M) = \max_{M' \in \mathcal{M}_{A,B}} w(M')$. Suppose that there exists $M_1 \in \mathcal{M}_{A,B}$ such that $w(M_1) > w(M)$. Notice that $B \cap M_1 = \emptyset$ and $A \subseteq M_1$, we have

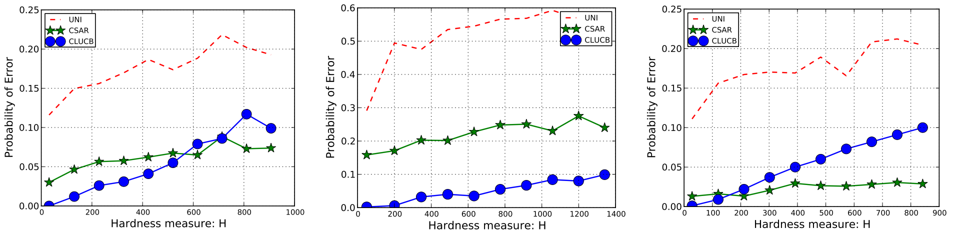
$$\begin{aligned} w_2(M_1) &= w_1(M_1) \\ &= w_1(M_1 \backslash A) + w_1(B) \\ &= w(M_1 \backslash A) + 3|A|L_1 \\ &= w(M_1) + 3|A|L_1 - w(A). \end{aligned}$$

Similarly, one can show that $w_2(M) = w(M) + 3|A|L_1 - w(A)$. By combining with the assumption that $w(M_1) > w(M)$ we see that $w_2(M_1) > w_2(M)$, which contradicts to the definition of M. Hence we have verified that $w(M) = \max_{M' \in \mathcal{M}_{A,B}} w(M')$. □

2.15 Preliminary Experiments: Identifying the Minimum Spanning Tree

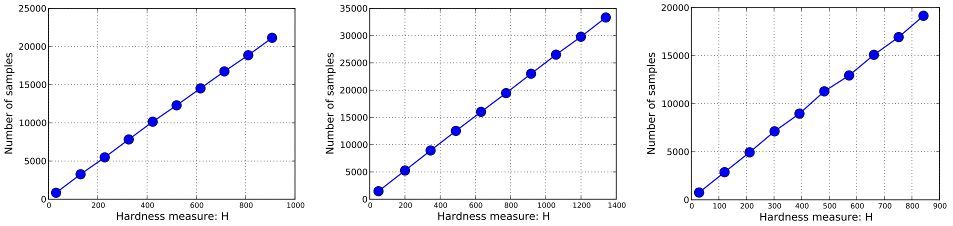
In this section, we present some preliminary experimental results of our algorithms CLUCB and CSAR. We conduct experiments on a real-world dataset with decision classes corresponding to spanning trees. We compare our algorithms with the uniform allocation benchmark UNI discussed in Section 2.12. The experiment results show that the proposed algorithms are considerably more sample efficient than the UNI algorithm, which agrees with our theoretical analysis.

Setup. Our task is to identify the optimal routing tree from a networking system which has the lowest expected latency in an exploration procedure, where one can obtain noisy measurements of latencies between different nodes. We model this problem as a CPE problem where the arms correspond to edges and the decision class corresponds to the set of spanning trees (which



(a) Network 1755 (b) Network 3257 (c) Network 3967

Figure 2.4: Comparison of empirical probability of errors with respect to **H**.

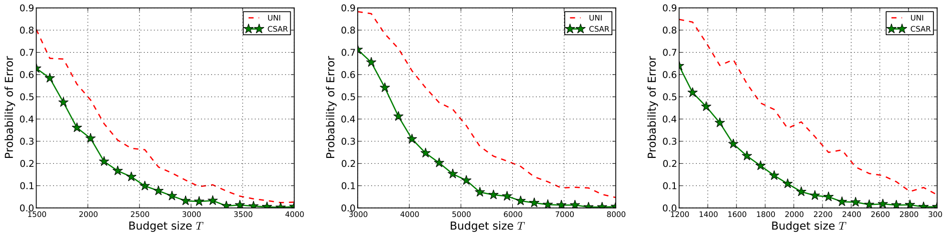


(a) Network 1755 (b) Network 3257 (c) Network 3967

Figure 2.5: Empirical sample complexity of CLUCB with respect to **H**.

is a special case of matroids, as we have discussed in Example 2.3). We use a real-world dataset called RocketFuel [140], which contains several ISP networks with routing information such as average latencies between nodes pairs. We select three medium-sized ISP networks with numbers of edges ranging from 161 to 328. For each network, we model the latency $X(e)$ of edge e as the sum of the given average latency $l(e)$ and an additive random noise $\mathcal{N}(0,1)$. Then we model the reward of edge e as the negative latency $-X(e)$ and therefore the expected reward of e is given by $w(e) = -l(e)$. Notice that we now need to find the spanning tree that maximizes the expected reward, which is exactly an instance of CPE.

Since the ground-truth of expected reward $\boldsymbol{w}$ is known, we can compute the ground-truth of the optimal set M_* and the hardness measures **H**. Furthermore, in order to investigate the relationship between **H** and sample complexity empirically, we generate a number of instances with different **H** by adjusting the



(a) Network 1755 ($\mathbf{H} = 117.6$) (b) Network 3257 ($\mathbf{H} = 181.5$) (c) Network 3967 ($\mathbf{H} = 108.7$)

Figure 2.6: Empirical probability of error of CSAR and UNI with respect to budget size T.

expected reward of each arm $e \in M_*$ with a same additive quantity c_0 while not changing the optimality of M_*. By definition of $\mathbf{H}$, we see that $\mathbf{H}$ decreases when c_0 increases.

Evaluation method. We use the following evaluation procedure to compare the sample efficiency between CLUCB,CSAR and UNI. Since CSAR and UNI are both learning algorithms in the fixed budget setting, the comparison between them is straightforward: for each given budget, we run both algorithms with this budget independently for 1000 times and compare their empirical probability of errors (the fraction of runs where a tested algorithm fails to report the ground-truth optimal set M_*). On the other hand, we use the following procedure to compare CLUCB with other fixed budget algorithms. For each instance of ISP network, we run CLUCB independently for 1000 times. Suppose that the i-th run of CLUCB uses T_i samples, we also run UNI and CSAR with budget T_i. Then we compare the empirical probability of errors of the tested algorithms after the 1000 runs are completed. In this way, we see that the compared algorithms use an equal number of samples in each run, which allows us to compare their sample efficiency. Finally, we set $\delta = 0.3$ for CLUCB throughout the experiments.

Experimental results. We test all competing algorithms using the aforementioned evaluation method. The experimental results are shown in Figure 2.4, Figure 2.5 and Figure 2.6. From the results (Figure 2.4 and Figure 2.6), we see that both CLUCB and CSAR are consistently more sample efficient than UNI by a large margin, i.e., they incur a smaller empirical probability of error than UNI when using a same number of samples. This matches our theoretical analyses of these algorithms. We also see that the probability of error of CLUCB is always smaller than the guarantee $\delta = 0.3$ (Figure 2.4) and the sample complexity of CLUCB is approximately linear in $\mathbf{H}$ (Figure 2.5), which agrees with our theory that the sample complexity bound for the spanning tree decision class is $\tilde{O}(\mathbf{H})$ (see Example 2.3).

Chapter 3

Linear Combinatorial Bandits

In this chapter, we apply combinatorial bandit models to recommender systems. In particular, we study the list-wise recommendation problem in a cold start setting, in which the objective is to recommend lists of items over a sequence of rounds to a new user with insufficient historical records.

We develop a model called *linear combinatorial bandit* in which a learning algorithm can dynamically identify diverse items that interest a new user. Specifically, each item is represented as a feature vector, and each user is represented as an unknown preference vector. At each of the T rounds, the algorithm sequentially selects a set of items according to the item-selection strategy that balances *exploration* and *exploitation*, and collects the user feedback on these selected items. A reward function is further designed to measure the quality (e.g. relevance or diversity) of the selected set based on observed feedback, and the goal of the algorithm is to maximize the total rewards of T rounds. The reward function only needs to satisfy two mild assumptions. that is general enough to accommodate a large class of nonlinear functions. In this chapter, we provide a learning algorithm that achieves $\tilde{O}(\sqrt{T})$ regret after playing T rounds. Experiments conducted on real-wold movie recommendation dataset demonstrate that our approach can effectively address the above challenges and hence improve the performance of recommendation task.

3.1 Introduction

Stochastic linear bandits are a natural extension of stochastic MABs, where arms are associated with feature vectors and rewards are determined by a linear function. Similar to the stochastic MABs, linear bandits naturally characterize the tradeoff between exploration and exploitation. Recently, researchers have applied linear bandit algorithms to develop recommender systems that can adapt to user feedback over time [93, 45].

However, in practice, a recommender system usually provides a set of movies, rather than a single movie, to a user at each time that the user visits the system. Using the terminology of MABs, this setting corresponds to the bandit game where a set of arms, instead of a single arm, are played at each round. In this setting, we need to extend the definition of reward from single arms to sets of arms. For example, in recommender systems, one may agree that a good definition of the reward of a set of movies should not simply be the sum of ratings of each movie in this set; it should also take diversity of the recommended set in to consider, in order to to avoid redundant or over-narrowed recommendation system. This "diversity promoting" reward function can be inherently non-linear. In order to model this sequential decision making problem, one would need a new model that (1) allows each arm to be associated with a feature vector; (2) allows a set of arms to be played at each round; and (3) allows the reward function to be non-linear (with respect to the rewards of each individual arms).

In this chapter, we develop such a model, which is called *linear combinatorial bandit*. Our model can be used to address recommendation problems emerged from real-world applications, where one need to recommend sets of diversified items. For this model, we develop LinCUCBand prove a $\tilde{O}(\sqrt{n})$ regret bound for the LinCUCB algorithm after playing n rounds[1]. We also develop Rarely Switching LinCUCB algorithm, which reduces the query complexity of LinCUCB algorithm at the cost of a slightly increased α-regret. We apply the framework to the diversified movie recommendation setting and conduct experiments on a public dataset.

3.2 Related Work

Most of traditional recommendation techniques focus on learning user preference according to users' historical records [88, 130, 29]. However, recent studies show that historical records may not well represent user interest [132, 154]. On the one hand, some users may not provide sufficient records, in which case it is crucial to predict user preference dynamically according to user feedback [103, 106]. On the other hand, users can hold diverse interest, and thus recommendation techniques should not only aim at increasing relevance, but also consider improving diversity of recommended results [154, 116].

To recommend items to users without sufficient historical records, several studies formulate this task as a multi-armed bandit problem [93, 94, 22]. Multi-armed bandit is a well-studied topic in the fields of statistics and machine learning (cf. [19, 5]). In traditional non-linear bandit problem, the learner cannot access the features of arms, and the rewards of different arms are independent. In this setting, the upper confidence bound (UCB) algorithm is proven to be theoretically optimal [91, 9]. However, without using arm features, the performance of UCB algorithm is quite limited in many practical scenarios, especially when there are a large number of arms [93]. On the other hand, linear bandit problem considers the case where the learner can observe the features of arms. Consequently, the learner can use these ob-

[1] $\tilde{O}(\cdot)$ is variant of big O notation that ignores logarithmic factors.

servations to infer the rewards of other unseen arms and improve the performance over time. Notably, Auer et al. [9] considered the linear bandit problem and developed LinRel algorithm that achieved an $\tilde{O}(\sqrt{n})$ regret bound after playing n rounds. Later, Li et al.[93] proposed LinUCB algorithm, which improves the practical performance of LinRel algorithm while enjoys similar regret bound [45]. They applied LinUCB algorithm on a personalized news recommendation task and demonstrated good performance [93].

In the settings of both non-linear and linear bandits, the learner is allowed to play one single arm on each round, i.e., recommend one item each time. However, recommending a single item on each round may not satisfy a user's diverse interest. Recently, several work generalized the classical non-linear bandits to combinatorial bandits [63, 62, 43], where the learner can play a set of arms, which is termed as a super arm, on each round. However, as generalizations of non-linear bandits, these work did not use arm features. Hence, their performance can be suboptimal in many recommendation tasks, particularly when the number of arms is large. Though our method inherits some concepts (e.g. super arm) from non-linear combinatorial bandit, both problem formulation and regret analysis are quite different, which are actually our main contributions.

Yue and Guestrin [155] proposed a linear submodular bandit approach for diversified retrieval. Their approach placed a strong restriction on user behavior. In particular, they assumed that user can only scan the items one by one in top-down fashion. In contrast, our framework has no limitation on user behavior. In addition, their framework is specifically designed for a certain type of submodular reward functions, while our approach allows a much larger class of reward functions.

3.3 Linear Combinatorial Bandit

In this section, we formulate the linear combinatorial bandit problem. Let n be the number of rounds and m be the number of arms. Let $\mathcal{S}_t \subseteq 2^{[m]}$ be the set of all possible subsets of arms on round t. We call each set of arms $S_t \in \mathcal{S}_t$ a *super arm*. At each round $t \in [n]$, a learner observes m feature vectors $\{\mathbf{x}_t(1), \ldots, \mathbf{x}_t(m)\} \subseteq \mathbb{R}^d$ corresponding to m arms. Then, the learner is asked to choose one super arm $S_t \in \mathcal{S}_t$ to play. Once a super arm $S_t \in \mathcal{S}_t$ is played, the learner observes the scores of arms in $\{r_t(i)\}_{i \in S_t}$ and receives a reward $R_t(S_t)$. For each arm $i \in [m]$, its score $r_t(i)$ is assumed to be

$$r_t(i) = \theta_*^T \mathbf{x}_t(i) + \epsilon_t(i), \tag{3.1}$$

where θ_* is a parameter unknown to the learner and the noise $\epsilon_t(i)$ is a zero-mean random variable. On the other hand, the reward $R_t(S_t)$ measures the quality of the super arm S_t and its definition will be specified later. The goal of the learner is to maximize the expected cumulative reward $\mathbb{E}\left[\sum_{t \in [n]} R_t(S_t)\right]$ over n rounds.

The reward $R_t(S_t)$ on round t is an application dependent function which measures the quality of recommended set of arms $S_t \subseteq [m]$. The reward can simply be the sum of the scores of arms in S_t, i.e. $R_t(S_t) = \sum_{i \in S_t} r_t(i)$. However, our framework also allows other more complicated non-linear rewards. For example, in addition to the sum of scores of arms, the reward $R_t(S_t)$ may also consider the “diversity” of arms in S_t, which can be defined as a non-linear function of features of arms.

Specifically, we consider the case where the expected reward $\mathbb{E}\left[R_t(S_t)\right]$ is a function of three variables: super arm S_t, feature vectors of arms $\mathbf{X}_t \triangleq \{\mathbf{x}_t(i)\}_{i \in [m]}$ and expected scores $\mathbf{r}_t^* \triangleq \{\theta_*^T \mathbf{x}_t(i)\}_{i \in [m]}$ associated with the arms. Formally, we denote the expected reward of playing S_t as $\mathbb{E}\left[R_t(S_t)\right] = f_{\mathbf{r}_t^*, \mathbf{X}_t}(S_t)$. By choosing different types of expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$, our framework covers both linear and non-linear rewards. Finally, in order to carry out our analysis, the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$ is required to satisfy the following assumptions.

Monotonicity The expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ is monotone non-decreasing with respect to the score vector $\mathbf{r}$. Formally, for any set of feature vectors of arms $\mathbf{X}$ and super arm S, if $r(i) \leq r'(i)$ for all $i \in [m]$, we have $f_{\mathbf{r},\mathbf{X}}(S) \leq f_{\mathbf{r}',\mathbf{X}}(S)$.

Lipschitz continuity The expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ is Lipschitz continuous with respect to the score vector $\mathbf{r}$ restricted on the arms in S. In particular, there exists a universal constant $C > 0$ such that, for any two score vectors $\mathbf{r}$ and $\mathbf{r}'$, we have $|f_{\mathbf{r},\mathbf{X}}(S) - f_{\mathbf{r}',\mathbf{X}}(S)| \leq C\sqrt{\sum_{i\in S}\left[r(i) - r'(i)\right]^2}$.

Our framework does not require the player to have direct knowledge on how the reward function $f_{\mathbf{r},\mathbf{X}}(S)$ is defined. Alternatively, we assume that the player has access to an oracle $\text{Oracle}_{\mathcal{S}}(\mathbf{r}, \mathbf{X})$, which takes the expected scores $\mathbf{r}$ and arms $\mathbf{X}$ as input, and returns the solution of the maximization problem $\arg\max_{S\in\mathcal{S}} f_{\mathbf{r},\mathbf{X}}(S)$. Since the maximization problems of many reward functions $f_{\mathbf{r},\mathbf{X}}(\cdot)$ of practical interest are NP-hard, our framework allows the oracle to produce an approximate solution to the problem. More precisely, an oracle $\text{Oracle}_{\mathcal{S}}(\mathbf{r}, \mathbf{X})$ is called *α-approximation oracle* for some $\alpha \leq 1$, if given input $\mathbf{r}$ and $\mathbf{X}$, the oracle always returns a super arm $S = \text{Oracle}_{\mathcal{S}}(\mathbf{r}, \mathbf{X}) \in \mathcal{S}$ satisfying $f_{\mathbf{r},\mathbf{X}}(S) \geq \alpha\,\text{opt}_{\mathbf{r},\mathbf{X}}$, where $\text{opt}_{\mathbf{r},\mathbf{X}} = \max_{S\in\mathcal{S}} f_{\mathbf{r},\mathbf{X}}(S)$ is the optimal value of the reward function. Under this setting, when $\alpha = 1$, the α-approximation oracle is exact and always produces the optimal solution.

Recall that the goal of the learner is to maximize its cumulative reward without knowing θ_*. Clearly, with the knowledge of θ_*, the optimal strategy is to choose $S_t = \arg\max_{S_t\in\mathcal{S}_t} f_{\mathbf{r}_t,\mathbf{X}_t}(S_t)$

on round t. Hence, it is natural to evaluate a learner relative to this optimal strategy and the difference of the learner's total reward and the total reward of the optimal strategy is called *regret*. However, if a learner only has accesses to an α-approximation oracle for some $\alpha < 1$, such evaluation would be unfair. Hence, in this chapter, we use the notion of α-regret which compares the learner's strategy with α-fraction of the optimal rewards on round t. Formally, the α-regret on round t can be written as

$$\mathrm{Reg}_t^\alpha = \alpha \operatorname*{opt}_{\mathbf{r}_t, \mathbf{X}_t} - f_{\mathbf{r}_t, \mathbf{X}_t}(S_t), \tag{3.2}$$

and we are interested in designing an algorithm whose total α-regret $\sum_{t=1}^T \mathrm{Reg}_t^\alpha$ is as small as possible.

3.4 Algorithm and α-Regret Analysis

In this section, we present linear combinatorial upper confidence bound algorithm (LinCUCB). LinCUCB is a general and efficient algorithm for the linear combinatorial bandit problem. The basic idea of LinCUCB is to maintain a confidence set for the true parameter θ_*. For each round t, the confidence set is constructed from feature vectors $\mathbf{X}_1, \ldots, \mathbf{X}_{t-1}$ and observed scores of selected arms $\{r_1(i)\}_{i \in S_1}, \ldots, \{r_{t-1}(i)\}_{i \in S_{t-1}}$ from previous rounds. As we will see later (Theorem 3.3), our construction of the confidence sets ensures that the true parameter θ_* lies in the confidence set with high probability. Using this confidence set of parameter θ_* and feature vectors of arms $\mathbf{X}_t$, the algorithm can efficiently compute an upper confidence bound for each score $\hat{\mathbf{r}}_t = \{\hat{r}_t(1), \ldots, \hat{r}_t(m)\}$. The upper confidence bounds $\hat{\mathbf{r}}_t$ and feature vectors of arms $\mathbf{X}_t$ are given to the oracle as input. Then, the algorithm plays the super arm returned by the oracle and uses the observed scores to adjust the confidence sets. The pseudocode of the algorithm is listed in Algorithm 5. The algorithm has time complexity $O(n(d^3 + md + h))$, where h denotes the time complexity of the oracle.

Algorithm 5 LinCUCB

```
**input:** $\lambda, \alpha_1, \dots, \alpha_n$
Initialize $\mathbf{V}_0 \leftarrow \lambda \mathbf{I}_{d \times d}, \mathbf{b}_0 \leftarrow \mathbf{0}_d$
**for** $t \leftarrow 1, \dots, n$ **do**
    $\hat{\theta}_t \leftarrow \mathbf{V}_{t-1}^{-1} \mathbf{b}_{t-1}$
    **for** $i \in 1, \dots, m$ **do**
        $\bar{r}_t(i) \leftarrow \hat{\theta}_t^T \mathbf{x}_t(i)$
        $\hat{r}_t(i) \leftarrow \bar{r}_t(i) + \alpha_t \sqrt{\mathbf{x}_t(i)^T \mathbf{V}_{t-1}^{-1} \mathbf{x}_t(i)}$
    **end for**
    $S_t \leftarrow \text{Oracle}_{\mathcal{S}_t}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$
    Play super arm $S_t$ and observe $\{r_t(i)\}_{i \in S_t}$
    $\mathbf{V}_t \leftarrow \mathbf{V}_{t-1} + \sum_{i \in S_t} \mathbf{x}_t(i) \mathbf{x}_t(i)^T$
    $\mathbf{b}_t \leftarrow \mathbf{b}_{t-1} + \sum_{i \in S_t} r_t(i) \mathbf{x}_t(i)$
**end for**
```

We now state our main theoretical result, a bound on the α-regret of Algorithm 5 when run with an α-approximation oracle. To carry out our analysis, we will need to assume that the l_2-norms of parameter θ_* and feature vectors of arms $\mathbf{X}_t$ are bounded. Using this assumption together with the monotonicity and Lipschitz continuity properties of the expected reward function $f_{\mathbf{r},\mathbf{X}}(\cdot)$, the following theorem states that the α-regret of Algorithm 5 is at most $O(d \log(n)\sqrt{n} + \sqrt{nd \log(n/\delta)})$, or $\tilde{O}(\sqrt{n})$ if one ignores logarithmic factors and regards the dimensionality of the parameter d as a constant.

Theorem 3.1. (α-regret bound of Algorithm 5). *Without loss of generality, assume that $\|\theta_*\|_2 \leq S$, $\|\mathbf{x}_t(i)\|_2 \leq 1$ and $r_t(i) \in [0, 1]$ for all $t \geq 0$ and $i \in [m]$. Given $0 < \delta < 1$, set $\alpha_t = \sqrt{d \log\left(\frac{1+tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$. Then, with probability at least $1 - \delta$, the total α-regret of LinCUCB algorithm satisfies*

$$\sum_{t=1}^{n} Reg_t^{\alpha} \leq C \sqrt{64 nd \log(1 + nm/d\lambda)} \cdot$$

$$\left(\sqrt{\lambda}S + \sqrt{2\log(1/\delta) + d\log(1 + nm/(\lambda d))}\right),$$

for any $n \geq 0$.

Note that the requirements $\|\mathbf{x}_t(i)\|_2 \leq 1$ and $r_t(i) \in [0,1]$ can be satisfied through proper rescaling on $\mathbf{x}_t(i)$ and θ_*.

Our next result is a problem-dependent α-regret bound of Algorithm 5. Let $B_t = \left\{S \mid S \in \mathcal{S}_t, f_{\mathbf{r}_t,\mathbf{X}_t}(S) < \alpha\,\mathrm{opt}_{\mathbf{r}_t,\mathbf{X}_t}\right\}$ denote the set of "bad" super arms on round t. We define the gap Δ_t at time t as follows

$$\Delta_t = \alpha \underset{\mathbf{r}_r,\mathbf{X}_t}{\mathrm{opt}} - \max_{S\in B_t} f_{\mathbf{r}_t,\mathbf{X}_t}(S). \tag{3.3}$$

Furthermore, we define $\bar{\Delta}_n = \min_{1\leq t\leq n} \Delta_t$. Then, the α-regret of Algorithm 5 can be bounded in terms of $\bar{\Delta}_n$ as follows.

Theorem 3.2. (Problem-dependent α-regret bound of Algorithm 5) *Under the same assumption as in theorem 3.1, with probability at least* $1-\delta$*, we have*

$$\sum_{t=1}^{n} Reg_t^{\alpha} \leq \frac{64C}{\bar{\Delta}_n} d\log(1 + nm/d\lambda)\cdot$$
$$\left(\sqrt{d\log\left((1 + nm/\lambda)/\delta\right)} + \lambda^{1/2}S\right)^2.$$

3.4.1 Proof

We begin with restating a concentration result from Abbasi-Yadkori et al. [1]. This result states that the true parameter θ_* lies within an ellipsoid centered at $\hat{\theta}_t$ simultaneously for all $t \in [n]$ with high probability.

Theorem 3.3. ([1, Theorem 2]) *Suppose the observed scores* $r_t(i)$ *are bounded in* $[0,1]$*. Assume that* $\|\theta_*\|_2 \leq S$ *and* $\|\mathbf{x}_t(i)\|_2 \leq 1$ *for all* $t \geq 0$ *and* $i \in [m]$*. Define* $\mathbf{V}_t = \mathbf{V} + \sum_{t=1}^{n}\sum_{i\in S_t} \mathbf{x}_t(i)\mathbf{x}_t(i)^T$

and set $\mathbf{V} = \lambda \mathbf{I}$*. Then, with probability at least* $1-\delta$*, for all round* $t \geq 0$*, the estimate* $\hat{\theta}_t$ *satisfies* [2]

$$\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \leq \sqrt{d \log\left(\frac{1 + tm/\lambda}{\delta}\right)} + \lambda^{1/2} S.$$

The proof of Theorem 3.3 is based on the theory of self-normalized processes. For an introduction to this theory, we refer interested readers to [113, 50].

Next, using Theorem 3.3, we show that with high probability, the upper confidence bounds of scores $\hat{\mathbf{r}}_t$ also do not deviate far from the true value of scores $\mathbf{r}_t^*$ for each round $t \in [n]$.

Lemma 3.1. *If we set* $\alpha_t = \sqrt{d \log\left(\frac{1+tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$*, with probability at least* $1 - \delta$*, we have*

$$0 \leq \hat{r}_t(i) - r_t^*(i) \leq 2\alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}},$$

holds simultaneously for any round $t \geq 0$ *and any arm* $i \in [m]$*.*

Proof. By Theorem 3.3, the random event

$$\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \leq \sqrt{d \log\left(\frac{1 + tm/\lambda}{\delta}\right)} + \lambda^{1/2} S$$

holds for all $t \in [n]$ simultaneously with probability at least $1-\delta$.

Now assume the above random event happens, by the definition of $\hat{r}_t(i)$, we have

$$\begin{aligned}
&\left|\hat{r}_t(i) - r_t^*(i)\right| \\
=& \left|\hat{\theta}_t^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} - \theta_*^T \mathbf{x}_t(i)\right| \\
\leq& \left|(\hat{\theta}_t - \theta_*)^T \mathbf{x}_t(i)\right| + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\leq& \left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}
\end{aligned}$$

[2] We denote $\|\mathbf{a}\|_{\mathbf{M}} \triangleq \sqrt{\mathbf{a}^T \mathbf{M} \mathbf{a}}$, where $\mathbf{a}$ is a vector and $\mathbf{M}$ is a positive definite matrix.

$$\leq 2\alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}.$$

On the other hand, we have

$$\begin{aligned}
&\hat{r}_t(i) - r_*(i) \\
=&\hat{\theta}_t^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} - \theta_*^T \mathbf{x}_t(i) \\
=&(\hat{\theta}_t - \theta_*)^T \mathbf{x}_t(i) + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\geq& -\left\|\hat{\theta}_t - \theta_*\right\|_{\mathbf{V}_{t-1}} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} \\
\geq& -\alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} + \alpha_t \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}} = 0.
\end{aligned}$$

□

To prove our main result Theorem 3.1, we need the following technical lemma.

Lemma 3.2. *Let $\mathbf{V} \in \mathbb{R}^{d\times d}$ be a positive definite matrix. For all $t = 1, 2, \ldots$, let S_t be a subset of $[m]$ of size less than or equal to k and define $\mathbf{V}_n = \mathbf{V} + \sum_{t=1}^n \sum_{i\in S_t} \mathbf{x}_t(i)\mathbf{x}_t(i)^T$.*

Then, if $\lambda \geq k$ and $\left\|\mathbf{x}_t(i)\right\|_2 \leq 1$ for all t and i, we have

$$\begin{aligned}
\sum_{t=1}^n \sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2 &\leq 2\log\det \mathbf{V}_n - \log\det \mathbf{V} \\
&\leq 2d\log((\mathrm{trace}(\mathbf{V}) + nk)/d) \\
&\qquad - 2\log\det \mathbf{V}.
\end{aligned}$$

Proof. We have

$$\begin{aligned}
&\det(\mathbf{V}_n) \\
=&\det\left(\mathbf{V}_{n-1} + \sum_{i\in S_n} \mathbf{x}_n(i)\mathbf{x}_n(i)^T\right) \\
=&\det(\mathbf{V}_{n-1})\det\left(\mathbf{I} + \sum_{i\in S_n} (\mathbf{V}_{n-1}^{-1/2}\mathbf{x}_n(i))(\mathbf{V}_{n-1}^{-1/2}\mathbf{x}_n(i))^T\right)
\end{aligned}$$

$$
\begin{aligned}
&= \det(\mathbf{V}_{n-1}) \det\left(\mathbf{I} + \sum_{i \in S_n} \|\mathbf{x}_n(i)\|^2_{\mathbf{V}_{t-1}^{-1}}\right) \\
&= \det(\mathbf{V}) \prod_{t=1}^{n} \left(1 + \sum_{i \in S_t} \|\mathbf{x}_t(i)\|^2_{\mathbf{V}_{t-1}^{-1}}\right).
\end{aligned}
$$

Now, using the fact that $u \leq 2\log(1+u)$ for any $u \in [0,1]$ and that $\|\mathbf{x}_t(i)\|^2_{\mathbf{V}_{t-1}^{-1}} \leq \|\mathbf{x}_t(i)\|^2 / \lambda_{\min}(\mathbf{V}_{t-1}) \leq 1/\lambda \leq 1/k$, we obtain

$$
\begin{aligned}
\sum_{t=1}^{n} \sum_{i \in S_t} \|\mathbf{x}_t(i)\|^2_{\mathbf{V}_{t-1}^{-1}} &\leq 2 \sum_{t=1}^{n} \log\left(1 + \sum_{i \in S_t} \|\mathbf{x}_t(i)\|^2_{\mathbf{V}_{t-1}^{-1}}\right) \\
&= 2 \log\det \mathbf{V}_n - 2\log\det \mathbf{V}.
\end{aligned}
$$

We remain to bound $\log\det \mathbf{V}_n$. Since $\|\mathbf{x}_n(i)\|_2 \leq 1$ and $|S_i| \leq k$ for all $i \in [n]$, the trace of $\mathbf{V}_n$ can be bounded by $\text{trace}(\mathbf{V}_n) \leq \text{trace}(\mathbf{V}) + nk$. Apply the Determinant-Trace Inequality [1, Lemma 10], we have

$$
\log\det(\mathbf{V}_n) \leq d \log((\text{trace}(\mathbf{V}) + nk)/d).
$$

□

Based on Lemma 3.1, Lemma 3.2 and the two assumptions on the expected reward, we are now ready to prove our main theorem.

Proof. (Theorem 3.1) By Lemma 3.1, we have $\hat{r}_t(i) \geq r^*_t(i)$ holds simultaneously for all $t \in [n]$ and $i \in [m]$ with probability at least $1-\delta$. Now, assume that this random event holds and apply the monotonicity property of the expected reward, for any super arm $S \in \mathcal{S}_t$, we have $f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S) \geq f_{\mathbf{r}^*_t, \mathbf{X}_t}(S)$.

Let $S_t \in \mathcal{S}_t$ be the super arm returned by the oracle $S_t = \text{Oracle}_{\mathcal{S}_t}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$ on round t. We now show that $f_{\hat{\mathbf{r}}_t, \mathbf{X}_t}(S_t) \geq \alpha\, \text{opt}_{\mathbf{r}^*_t, \mathbf{X}_t}$. To see this, we denote $S^*_t = \arg\max_{S \in \mathcal{S}_t} f_{\mathbf{r}^*_t, \mathbf{X}_t}(S)$

as the maximizer of $f_{\mathbf{r}_t^*,\mathbf{X}_t}(\cdot)$ and $\hat{S}_t$ as the optimal solution of $\arg\max_{S\in\mathcal{S}_t} f_{\hat{\mathbf{r}}_t,\mathbf{X}_t}(S)$. Then, we have

$$\begin{aligned} f_{\hat{\mathbf{r}}_t,\mathbf{X}_t}(S_t) &\geq \alpha \underset{\hat{\mathbf{r}}_t,\mathbf{X}_t}{\mathrm{opt}} \\ &= \alpha f_{\hat{\mathbf{r}}_t,\mathbf{X}_t}(\hat{S}_t) \geq \alpha f_{\hat{\mathbf{r}}_t,\mathbf{X}_t}(S_t^*) \\ &\geq \alpha f_{\mathbf{r}_t^*,\mathbf{X}_t}(S_t^*) = \alpha \underset{\mathbf{r}_t^*,\mathbf{X}_t}{\mathrm{opt}}\,, \end{aligned}$$

where we have used the definition of α-approximation oracle and the optimality of $\hat{S}_t$.

Now, we can bound α-regret at round t as follows,

$$\begin{aligned} \mathrm{Reg}_t^\alpha &= \alpha \underset{\mathbf{r}_t^*,\mathbf{X}_t}{\mathrm{opt}} - f_{\mathbf{r}_t^*,\mathbf{X}_t}(S_t) \\ &\leq f_{\hat{\mathbf{r}}_t,\mathbf{X}_t}(S_t) - f_{\mathbf{r}_t^*,\mathbf{X}_t}(S_t) \\ &\leq C\sqrt{\sum_{i\in S_t}(\hat{r}_t(i) - r_t^*(i))^2} \\ &\leq C\sqrt{\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2}, \end{aligned}$$

where the second inequality follows from the Lipschitz continuity property of the expected reward $f_{\mathbf{r},\mathbf{X}}(\cdot)$.

Therefore, with probability at least $1-\delta$, for all $n \geq 0$,

$$\begin{aligned} \sum_{t=1}^n \mathrm{Reg}_t^\alpha &\leq \sqrt{n\sum_{t=1}^n (\mathrm{Reg}_t^a)^2} \\ &\leq C\sqrt{8n\sum_{t=1}^n\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2} \\ &\leq C\alpha_n\sqrt{32n}\sqrt{\sum_{t=1}^n\sum_{i\in S_t} \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2} \\ &\leq C\alpha_n\sqrt{32n}\sqrt{2d\log(\lambda + nm/d) - 2d\log\lambda} \end{aligned}$$

$$\leq C\sqrt{64nd\log(1+nm/d\lambda)}\cdot \left(\sqrt{d\log\left((1+nm/\lambda)/\delta\right)}+\sqrt{\lambda}S\right),$$

where the last inequality follows from Lemma 3.2, the fact that $|S_t| \leq m$ for all t and that $\mathbf{V} = \lambda\mathbf{I}$.

□

Proof. (Theorem 3.2) First we assume that, for all $t \in [n]$ and $i \in [n]$, we have $\hat{r}_t(i) \geq r_t^*(i)$. By Lemma 3.1, we see that this assumption holds with probability at least $1-\delta$.

Using an argument similar to the proof of Theorem 3.1, we know that, under our assumption, Reg_t^α is bounded by

$$\mathrm{Reg}_t^\alpha \leq C\sqrt{\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}}. \tag{3.4}$$

Next, since either $\mathrm{Reg}_t^\alpha \leq 0$ (this correspond to the case of choosing a super-arm S_t with reward not smaller than $\alpha\,\mathrm{opt}_{\mathbf{r}_t,\mathbf{X}_t}$) or $\mathrm{Reg}_t^\alpha \geq \Delta_t$, therefore we have

$$\mathrm{Reg}_t^\alpha \leq \frac{\mathrm{Reg}_t^{\alpha 2}}{\Delta_t}. \tag{3.5}$$

Hence, we have

$$\begin{aligned}\sum_{t=1}^n \mathrm{Reg}_t^\alpha &\leq \sum_{t=1}^n \frac{(\mathrm{Reg}_t^\alpha)^2}{\Delta_t}\\ &\leq \frac{32C}{\bar{\Delta}_n}\left(\sum_{t=1}^n\sum_{i\in S_t}\alpha_t^2\left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}\right)\\ &\leq \frac{32\alpha_n^2 C}{\bar{\Delta}_n}\left(\sum_{t=1}^n\sum_{i\in S_t}\left\|\mathbf{x}_t(i)\right\|^2_{\mathbf{V}_{t-1}^{-1}}\right)\\ &\leq \frac{32\alpha_n^2 C}{\bar{\Delta}_n}\left(2d\log(\lambda+nm/d)-2d\log\lambda\right)\end{aligned}$$

$$\leq \frac{64C}{\bar{\Delta}_n} d \log(1 + nm/d\lambda) \cdot \\ \left(\sqrt{d \log\left((1 + nm/\lambda)/\delta\right)} + \lambda^{1/2} S\right)^2,$$

where the first inequality follows from Eq. (3.5); the second inequality follows from Eq. (3.4); and the last inequality follows from Lemma 3.2 with the fact that $|S_t| \leq m$ for all t and $\mathbf{V} = \lambda \mathbf{I}$.

□

3.4.2 Reducing Query Complexity of LinCUCB

The proposed LinCUCB algorithm is a general learning algorithm for the linear combinatorial bandit problem. However, one potential drawback of LinCUCB is that it has a relatively high *query complexity*, i.e. it may queries the oracle for $O(n)$ time. This is important if making queries to the oracle is expensive. For example, the oracle may have a high computational complexity, or the oracle may cost certain amount of resources per query. It is clear that reducing the query complexity of LinCUCB would be beneficial for these cases.

In this part, we show that a modified LinCUCB algorithm need to call the oracle for only $O(\log(n))$ times, which leads to an exponential saving in the query complexity, with a small price on the α-regret. The idea is to re-query the oracle only when $\det(\mathbf{V}_t)$ is increased by a factor of $1 + \gamma$. Following the terminology of Abbasi-Yadkori et al. [1], we refer this modified LinCUCB algorithm as Rarely Switching LinCUCB algorithm. The pseudo-code of this algorithm is shown in Algorithm 6.

Our next result shows that the modified algorithm increases the regret by only a factor of $O(\sqrt{1+\gamma})$.

Theorem 3.4. (α-regret bound of Algorithm 6). *Use the same assumption as in Theorem 3.1. Assume that there exists* $\mathbf{X}$ *and* $\mathcal{S}$ *such that* $\mathbf{X} = \mathbf{X}_1 = \ldots = \mathbf{X}_n$ *and* $\mathcal{S} = \mathcal{S}_1 = \ldots = \mathcal{S}_n$. *Given* *parameter $\gamma > 0$, then, with probability at least $1 - \delta$, the total α-regret of Algorithm 6 satisfies*

$$\sum_{t=1}^{n} Reg_t^{\alpha} \le C\sqrt{1+\gamma}\sqrt{64nd\log(1+nm/d\lambda)} \cdot \left(\sqrt{\lambda}S + \sqrt{2\log(1/\delta) + d\log(1+nm/(\lambda d))}\right),$$

for any $n \ge 0$.

Algorithm 6 Rarely Switching LinCUCB

```
input: $\lambda, \alpha_1, \dots, \alpha_n, \gamma$
$\mathbf{V}_0 \leftarrow \lambda \mathbf{I}_{d\times d}, \mathbf{b}_0 \leftarrow \mathbf{0}_d$
$\mathbf{V}_{-1} \leftarrow \mathbf{0}_{d\times d}, \tau_0 \leftarrow 0$
for $t \leftarrow 1, \dots, n$ do
    if $\det(\mathbf{V}_{t-1}) \ge (1+\gamma)\det(\mathbf{V}_{\tau_{t-1}-1})$ then
        $\tau_t \leftarrow t$
        $\hat{\theta}_t \leftarrow \mathbf{V}_{t-1}^{-1}\mathbf{b}_{t-1}$
        for $i \in 1, \dots, m$ do
            $\bar{r}_t(i) \leftarrow \hat{\theta}_t^T \mathbf{x}(i)$
            $\hat{r}_t(i) \leftarrow \bar{r}_t(i) + \alpha_t\sqrt{\mathbf{x}(i)^T \mathbf{V}_{t-1}^{-1}\mathbf{x}(i)}$
        end for
        $S_t \leftarrow \mathrm{Oracle}_{\mathcal{S}_t}(\hat{\mathbf{r}}_t, \mathbf{X})$
    else
        $\tau_t \leftarrow \tau_{t-1}$
        $S_t \leftarrow S_{\tau_t}$
    end if
    Play super arm $S_t$ and observe $\{r_t(i)\}_{i\in S_t}$
    $\mathbf{V}_t \leftarrow \mathbf{V}_{t-1} + \sum_{i\in S_t}\mathbf{x}(i)\mathbf{x}(i)^T$
    $\mathbf{b}_t \leftarrow \mathbf{b}_{t-1} + \sum_{i\in S_t} r_t(i)\mathbf{x}(i)$
end for
```

We first prove a technical lemma in the following.

Lemma 3.3. *Let $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{d\times d}$ be three symmetric positive semi-definite matrices. Assume that $\mathbf{A} = \mathbf{B} - \mathbf{C}$. Then, we have*

$$\sup_{\mathbf{x}\neq 0} \frac{\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \le \frac{\det(\mathbf{B})}{\det(\mathbf{A})}.$$

Proof. First, similar to the proof of [1, Lemma 12], let us assume that $\mathbf{C} = \mathbf{v}\mathbf{v}^T$ for some $\mathbf{v} \in \mathbb{R}^d$. We first prove the claim for this special case. We will prove the general case later.

Fix an arbitrary non-zero vector $\mathbf{x} \in \mathbb{R}^d$. Then, we have

$$\begin{aligned}
\frac{\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} &= \frac{\mathbf{x}^T(\mathbf{B} - \mathbf{v}\mathbf{v}^T)^{-1}\mathbf{x}}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \\
&\overset{(a)}{=} \frac{\mathbf{x}^T\Big(\mathbf{B}^{-1} + \mathbf{B}^{-1}\mathbf{v}(1 - \mathbf{v}^T\mathbf{B}^{-1}\mathbf{v})^{-1}\mathbf{v}^T\mathbf{B}^{-1}\Big)\mathbf{x}}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \\
&= 1 + \frac{1}{1 - \mathbf{v}^T\mathbf{B}^{-1}\mathbf{v}} \cdot \frac{(\mathbf{x}^T\mathbf{B}^{-1}\mathbf{v})^2}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \\
&= 1 + \frac{1}{1 - \mathbf{v}^T\mathbf{B}^{-1}\mathbf{v}} \cdot \frac{(\mathbf{x}^T\mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{v})^2}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \\
&\overset{(b)}{\leq} 1 + \frac{1}{1 - \mathbf{v}^T\mathbf{B}^{-1}\mathbf{v}} \cdot \frac{\left\|\mathbf{x}^T\mathbf{B}^{-1/2}\right\|^2 \left\|\mathbf{B}^{-1/2}\mathbf{v}\right\|^2}{\mathbf{x}^T\mathbf{B}^{-1}\mathbf{x}} \\
&= 1 + \frac{1}{1 - \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2} \cdot \frac{\|\mathbf{x}\|_{\mathbf{B}^{-1}}^2 \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2}{\|\mathbf{x}\|_{\mathbf{B}^{-1}}^2} \\
&= 1 + \frac{\|\mathbf{v}\|_{\mathbf{B}^{-1}}^2}{1 - \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2} \\
&= \frac{1}{1 - \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2},
\end{aligned} \tag{3.6}$$

where (a) follows from Woodbury matrix identity and (b) follows from Cauchy-Schwartz inequality.

On the other hand, we have

$$\begin{aligned}
\det(\mathbf{A}) &= \det(\mathbf{B} - \mathbf{v}\mathbf{v}^T) \\
&= \det(\mathbf{B})\det(\mathbf{I} - (\mathbf{B}^{-1/2}\mathbf{v})(\mathbf{B}^{-1/2}\mathbf{v})^T) \\
&= \det(\mathbf{B})(1 - \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2).
\end{aligned}$$

Therefore we obtain

$$\frac{\det(\mathbf{B})}{\det(\mathbf{A})} = \frac{1}{1 - \|\mathbf{v}\|_{\mathbf{B}^{-1}}^2}. \tag{3.7}$$

By combining Eq. (3.6) and Eq. (3.7), we see that our claim holds for this case.

Next, we prove the claim for general $\mathbf{C}$. Since $\mathbf{C}$ is a symmetric positive semidefinite matrix, therefore there exists a sequence of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r \in \mathbb{R}^d$ such that $\mathbf{C} = \sum_{i=1}^r \mathbf{v}_i \mathbf{v}_i^T$.

Now, for each $i \in \{0, \ldots, r\}$, define $\mathbf{A}_i = \mathbf{B} - \left(\sum_{j=1}^i \mathbf{v}_j \mathbf{v}_j^T\right)$. Again fix an arbitrary nonzero vector $\mathbf{x} \in \mathbb{R}^d$. Notice that $\mathbf{A} = \mathbf{A}_r$ and $\mathbf{B} = \mathbf{A}_0$, we have

$$\begin{aligned}\frac{\mathbf{x}^T \mathbf{A}^{-1} \mathbf{x}}{\mathbf{x}^T \mathbf{B}^{-1} \mathbf{x}} &= \frac{\mathbf{x}^T \mathbf{A}_r^{-1} \mathbf{x}}{\mathbf{x}^T \mathbf{A}_{r-1}^{-1} \mathbf{x}} \cdot \frac{\mathbf{x}^T \mathbf{A}_{r-1}^{-1} \mathbf{x}}{\mathbf{x}^T \mathbf{A}_{r-2}^{-1} \mathbf{x}} \cdots \frac{\mathbf{x}^T \mathbf{A}_1^{-1} \mathbf{x}}{\mathbf{x}^T \mathbf{A}_0^{-1} \mathbf{x}} \\ &\leq \frac{\det(\mathbf{A}_{r-1})}{\det(\mathbf{A}_r)} \cdot \frac{\det(\mathbf{A}_{r-2})}{\det(\mathbf{A}_{r-1})} \cdots \frac{\det(\mathbf{A}_0)}{\det(\mathbf{A}_1)} \\ &= \frac{\det(\mathbf{B})}{\det(\mathbf{A})},\end{aligned}$$

where the inequality follows from the claim proved in the first part.

□

Then, we are ready to prove Theorem 3.4 by combining the argument for Theorem 3.1 and Lemma 3.3.

Proof. (Theorem 3.4) By Theorem 3.3 and the definition of α_t, we see that the random event

$$\left\{\forall t \in [n], \quad \|\theta_t - \theta_*\|_{\mathbf{V}_{t-1}} \leq \alpha_t\right\}$$

holds with probability at least $1 - \delta$. We shall assume that this random event holds in the rest of the proof.

For all $t \in [n]$, we see that τ_t is the smallest time such that $S_t = S_{\tau_t}$. Let $i \in [m]$ be an arbitrary arm and let $\mathbf{x}(i)$ be the feature vector of the i-th arm.

Applying Lemma 3.3, we obtain

$$\|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} = \sqrt{\mathbf{x}(i)^T \mathbf{V}_{\tau_t-1}^{-1} \mathbf{x}(i)}$$

$$\begin{aligned}
&\leq \sqrt{\left(\mathbf{x}(i)^T \mathbf{V}_t^{-1} \mathbf{x}(i)\right)} \sqrt{\frac{\det(\mathbf{V}_{t-1})}{\det(\mathbf{V}_{\tau_t - 1})}} \\
&= \|\mathbf{x}(i)\|_{\mathbf{V}_{t-1}^{-1}} \sqrt{\frac{\det(\mathbf{V}_{t-1})}{\det(\mathbf{V}_{\tau_t - 1})}} \\
&\leq \sqrt{1+\gamma}\, \|\mathbf{x}(i)\|_{\mathbf{V}_{t-1}^{-1}} . \qquad (3.8)
\end{aligned}$$

Next, we bound $|\hat{r}_{\tau_t}(i) - r^*(i)|$ as follows.

$$\begin{aligned}
|\hat{r}_{\tau_t}(i) - r^*(i)| &= \left|\hat{\theta}_{\tau_t}^T \mathbf{x}(i) + \alpha_{\tau_t} \|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} - \theta_*^T \mathbf{x}(i)\right| \\
&\leq \left|\hat{\theta}_{\tau_t}^T \mathbf{x}(i) - \theta_*^T \mathbf{x}(i)\right| + \alpha_{\tau_t} \|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} \\
&\leq \left\|\hat{\theta}_{\tau_t} - \theta_*\right\|_{\mathbf{V}_{\tau_t-1}} \|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} \\
&\qquad + \alpha_{\tau_t} \|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} \\
&\leq 2\alpha_{\tau_t} \|\mathbf{x}(i)\|_{\mathbf{V}_{\tau_t-1}^{-1}} .
\end{aligned}$$

One can also show that

$$\hat{r}_{\tau_t}(i) \geq r^*(i).$$

Define $S^* = \arg\max_{S \in \mathcal{S}} f_{\mathbf{r}^*,\mathbf{X}}(S)$ and $\hat{S}_{\tau_t} = \arg\max_{S \in \mathcal{S}} f_{\hat{\mathbf{r}}_{\tau_t},\mathbf{X}}(S)$. Then, we have

$$\begin{aligned}
f_{\hat{\mathbf{r}_{\tau_t}},\mathbf{X}}(S_{\tau_t}) &\geq \alpha \operatorname*{opt}_{\hat{\mathbf{r}_{\tau_t}},\mathbf{X}} \\
&= \alpha f_{\hat{\mathbf{r}_{\tau_t}},\mathbf{X}}(\hat{S}_{\tau_t}) \geq \alpha f_{\hat{\mathbf{r}_{\tau_t}},\mathbf{X}}(S^*) \\
&\geq \alpha f_{\mathbf{r}^*,\mathbf{X}}(S^*) = \alpha \operatorname*{opt}_{\mathbf{r}^*,\mathbf{X}},
\end{aligned}$$

where we have used again the definition of α-approximation oracle.

Now we can bound the regret on time t as follows

$$\mathrm{Reg}_t^{\alpha} = \alpha \operatorname*{opt}_{\mathbf{r}^*,\mathbf{X}} - f_{\mathbf{r}^*,\mathbf{X}}(S_t)$$

$$
\begin{aligned}
&= \alpha \operatorname*{opt}_{\mathbf{r}^*,\mathbf{X}} - f_{\mathbf{r}^*,\mathbf{X}}(S_{\tau_t}) \\
&\leq f_{\hat{\mathbf{r}_{\tau_t}},\mathbf{X}}(S_{\tau_t}) - f_{\mathbf{r}^*,\mathbf{X}}(S_{\tau_t}) \\
&\leq C\sqrt{\sum_{i\in S_{\tau_t}} (\hat{r}_{\tau_t}(i) - r^*(i))^2} \\
&\leq C\sqrt{\sum_{i\in S_{\tau_t}} 4\alpha_{\tau_t}^2 \left\|\mathbf{x}(i)\right\|_{\mathbf{V}_{\tau_t-1}^{-1}}^2} \\
&\leq C\sqrt{1+\gamma}\sqrt{\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2}, \qquad (3.9)
\end{aligned}
$$

where Eq. (3.9) follows from Eq. (3.8).

Therefore, by summing up the regret for all round $t \in [n]$, we obtain

$$
\begin{aligned}
\sum_{t=1}^{n} \mathrm{Reg}_t^\alpha \leq & \sqrt{n\sum_{t=1}^{n}(\mathrm{Reg}_t^a)^2} \\
&\leq C\sqrt{1+\gamma}\sqrt{8n\sum_{t=1}^{n}\sum_{i\in S_t} 4\alpha_t^2 \left\|\mathbf{x}_t(i)\right\|_{\mathbf{V}_{t-1}^{-1}}^2} \\
&\leq C\sqrt{1+\gamma}\sqrt{64nd\log(1+nm/d\lambda)}\cdot \\
&\quad \left(\sqrt{d\log\left((1+nm/\lambda)/\delta\right)} + \sqrt{\lambda}S\right),
\end{aligned}
$$

where the second inequality follows from Eq. (3.9) and the rest of derivation is same to the proof of Theorem 3.1.

□

3.5 Application to Online Diversified Recommendation

In this section, we apply our algorithm to diversified movie recommendation tasks. In this application, the recommender system recommends sets of movies, rather than individual ones. In addition, the recommended movies should be diversified such that the coverage of information that interests users is maximized. Furthermore, the recommender system need to use the user's feedback to improve its performance for future recommendations.

This application can be naturally formulated as a linear combinatorial problem as follows. Suppose, on each round t, there are m available movies and each movie is represented as a feature vector $\mathbf{x}_t(i) \in \mathbb{R}^d$. We can view the m movies as m arms and regard the feature vectors of movies as the feature vectors associated with arms. Then, the parameter $\theta_* \in \mathbb{R}^d$ corresponds to the user's (unknown) preference and the scores $r_t(i)$ are the ratings given by the user. At each round t, the system need to recommend a set of exactly k movies. This cardinality constraint is equivalent to assign the set of allowed super arms $\mathcal{S}_t = \{S | S \in 2^{[m]} \text{ and } |S| = k\}$ to be the set of all subsets of size k for all $t \geq 0$.

Next, we define the expected reward $f_{\mathbf{r},\mathbf{X}}(S)$ of a super arm S and construct an α-approximation oracle that associates to the expected reward. The definition of reward of super arm S should reflect both relevance and diversity of the set of movies in the super arm. We consider the following definition of reward which is proposed recently by Qin and Zhu [116],

$$f_{\mathbf{r},\mathbf{X}}(S) = \sum_{i \in S} r(i) + \lambda h(S, \mathbf{X}), \tag{3.10}$$

where $h(S, \mathbf{X}) = \frac{1}{2}|S| \log(2\pi e) + \frac{1}{2} \log\det(\mathbf{X}(S)^T \mathbf{X}(S) + \sigma^2 \mathbf{I})$ is called entropy regularizer since it quantifies the posterior uncertainty of ratings of movies in the set S. Here, the matrix $\mathbf{X}(S) \in \mathbb{R}^{d \times |S|}$ denotes a submatrix of $\mathbf{X}$ that consists of columns indexed by S and σ^2 is a smoothing parameter. This definition of entropy regularizer is derived as the differential entropy of ratings based on the Probabilistic Matrix Factorization (PMF) model. The derivation is omitted and we refer interested readers to [116] for details. Finally, the parameter λ of Eq. (3.10) is a regularization constant which trades-off between relevance and diversity.

As shown in [116], a simple greedy algorithm is guaranteed to find the super arm with reward larger than $(1 - 1/e)OPT$, where OPT is the reward of the best super arm [110, 116]. By definition, this algorithm can be employed as a valid $(1 - 1/e)$-approximation oracle in the linear combinatorial bandit framework. We denote this oracle by $\mathcal{O}_k^{\text{div}}(\mathbf{r}, \mathbf{X})$.

By plugging this oracle $\mathcal{O}_k^{\text{div}}(\mathbf{r}, \mathbf{X})$ in LinCUCB , we can construct an algorithm for the online diversified movie recommendation application. This can be done by simply changing Line 9 of Algorithm 5 to $S_t \leftarrow \mathcal{O}_k^{\text{div}}(\hat{\mathbf{r}}_t, \mathbf{X}_t)$. We denote the resulting algorithm as LinCUCB$^{\text{div}}$. This algorithm enjoys the same regret bound we established in Theorem 3.1. We refer interested readers to [117] for details of the LinCUCB$^{\text{div}}$algorithm.

3.6 Experiments

3.6.1 Experiment Setup

We conduct experiments on the MovieLens dataset, which is a public dataset consisting 1,000,029 ratings for 3900 movies by 6040 users of online movie recommendation service [108]. Each element of the dataset is represented by a tuple $t_{i,j} = (u_i, v_j, r_{i,j})$, where u_i denotes userID, v_j denotes movieID, and $r_{i,j}$ which is an integer score between 1 and 5 denotes the rating of user i for movie j (higher score indicates higher preference).

We split the dataset into training and test set as follows. We construct the test set by randomly selecting 300 users such that each selected user has at least 100 ratings. The remaining 5740 users and their ratings belong to the training set. Then, we apply a rank-d probabilistic matrix factorization (PMF) algorithm on the training data to learn the feature vectors of movies (each feature vector is d-dimensional). These feature vectors will be used later by the bandit algorithms as the feature vectors of arms.

Baselines. We compare our combinatorial linear bandit algorithm with the following baselines.

k-LinUCB algorithm. LinUCB [93] is a linear bandit algorithm which recommends exactly one arm at each time. To recommend a set of k movies, we repeat LinUCB algorithm k times on each round. By sequentially removing recommended arms, we ensure the k arms returned by LinUCB are distinct on each round. Finally, we highlight that the resulting bandit algorithm can be regarded as a combinatorial linear bandit with linear expected reward function

$$f_{\mathbf{r},\mathbf{X}}(S) = \sum_{i \in S} r(i).$$

Therefore, the major difference between k-LinUCB algorithm and our LinCUCB$^{\text{div}}$algorithm, which uses a reward function defined in Eq. (3.10), lies in that our algorithm optimizes the diversity of arms in set S_t.

Warm-start diversified movie recommendation. We denote this baseline as “warm-start” for short. For each user u in test set, we randomly select η ratings to train an user preference vector using PMF model. We call the parameter η as warm-start offset. With the estimated preference vector, one can repeatedly recommend sets of diverse recommendation results by maximizing the reward function 3.10. Note that this method cannot dynamically adapts to user’s feedback, and thus each round is independent with the others.

Metric. We use precision to measure the quality of recommended movie sets over n rounds. Specifically, for each user u

in the test test, we define the set of "preferred movies" L_u as the set of movies which user u assigned a rating of 4 or 5. Intuitively, a good movie set recommendation algorithm should recommend movie sets which cover a large fraction of preferred movies. Formally, on round t, suppose the recommendation algorithm recommends a set of movies S_t. The precision $p_{t,u}$ of user u on round t is defined as

$$p_{t,u} = \frac{|S_t \cap L_u|}{|S_t|}.$$

Then, the average precision of P_t of all test users up to round t is given by

$$P_t = \frac{1}{t|U|} \sum_{u \in U} \sum_{i=1}^{t} p_{i,u}.$$

Note that we do not aim at predicting the ratings of movies, but to provide more satisfying recommendation lists. Hence, precision is a more appropriate metric rather than the root mean square error (RMSE).

Actually, our algorithm (as well as baselines) essentially used an l_2-regularized linear regression method to predict movie ratings based on existing observations. This is equivalent to the rating prediction methods used by many matrix factorization algorithms, which are shown to have low RMSEs [107]. Moreover, we cannot use regret as a metric either, because the definitions of regrets vary greatly for different bandit algorithms.

3.6.2 Experiment Results

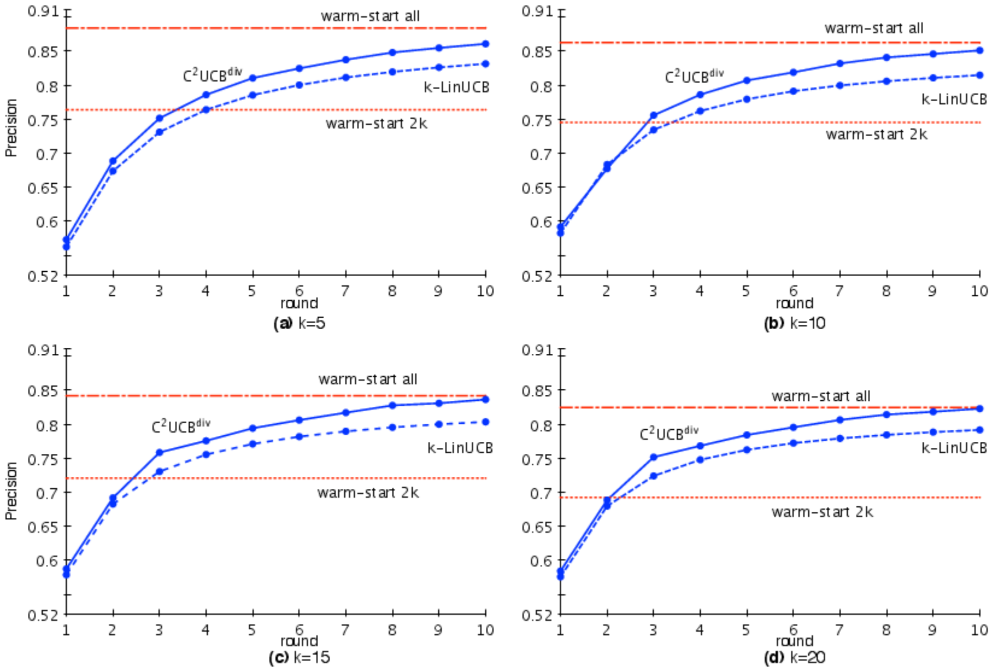
We consider recommending different number of movies to each user on each round, i.e., the size of super arm k takes values in $\{5, 10, 15, 20\}$. For each k, we set the exploration parameter $\alpha_t = 1.0$. The parameters of entropy regularizer are set to be $\lambda = 0.5$ and $\sigma = 1.0$.

	$t=5$		$t=10$		warm-start	
	KL	CC	KL	CC	$\eta=2k$	$\eta=\text{all}$
$k=5$	0.785	0.810	0.831	0.861	0.763	0.884
$k=10$	0.779	0.806	0.814	0.851	0.745	0.862
$k=15$	0.770	0.793	0.803	0.836	0.720	0.841
$k=20$	0.762	0.784	0.791	0.822	0.692	0.824

Table 3.1: Precision values of competing algorithms. CC: LinCUCB$^{\text{div}}$algorithm. KL: k-LinUCB algorithm.

For the warm-start baseline that allows an offline-estimated user preference, we consider two cases where η takes different values. In one case that we denote as "warm-start 2k", $\eta = k \times 2$ which indicates the method can access ratings of two rounds. In the other case that we denote as "warm-start all", η equals the total amount of observations, which indeed corresponds to the best solution, i.e., all observations are used to train user preference.

The results are shown in Figure 3.1 and Table 3.1, where our approach is denoted as LinCUCB$^{\text{div}}$. We can see that, in all cases, the "warm-start 2k" baseline outperforms bandit algorithms on earlier rounds, which is reasonable since the "warm-start 2k" baseline is provided warm-start observations to learn the user preference. But when more user feedbacks are available, bandit algorithms improve performance by dynamically adapting to user feedbacks. Near the end of 10 rounds, LinCUCB$^{\text{div}}$ can achieve a result that is comparable to "warm-start all". Compared to k-LinUCB, our method LinCUCB$^{\text{div}}$ finds a better match between recommended movies and user interest (i.e., the movies liked by each given user), and thus improves the overall performance. Furthermore, when k is larger, LinCUCB$^{\text{div}}$ algorithm obtains larger performance gain over k-LinUCB algorithm. The experiment results indicate that our method helps uncover users' diverse interest by using a non-linear diversity promoting reward function.



(a) k=5

(b) k=10

(c) k=15

(d) k=20

Figure 3.1: Experiment results comparing LinCUCB$^{\text{div}}$ with k-LinUCB, warm-start $2t$ and warm-start all on different choices of k.

3.7 Conclusion

We presented a general model called linear combinatorial bandit that accommodates combinatorial nature of linear arms. We developed an efficient algorithm LinCUCB for linear combinatorial bandit and provide a rigorously regret analysis. We further applied this model on online diversified movie recommendation task, and developed a specific algorithm LinCUCB$^{\mathrm{div}}$for this application. Experiments on public MovieLens dataset demonstrate that our approach helps explore and exploit users' diverse preference, and hence improves the performance of recommendation task.

Chapter 4

Fast Relative-Error Approximation Algorithm for Ridge Regression

A key component used in the learning algorithms of linear stochastic bandits is ridge regression [93, 45, 1], which is also widely applied in other areas of machine learning and statistics. In many of these applications, the number of features p is often much larger than the number of samples n. When $p > n$, the popular solution of ridge regression takes $O(n^2p + n^3)$ time. It is clear that this is expensive for large datasets where $p \gg n \gg 1$.

In this chapter, we present an $o(n^2p)$ time approximation algorithm for ridge regression with a provable relative error bound. Specifically, assuming that the data matrix $\mathbf{A}$ is full-rank, our algorithm computes an approximate solution $\tilde{\mathbf{x}}$ satisfying that $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2$ with high probability, and runs in $\tilde{O}(\mathrm{nnz}(\mathbf{A}) + n^3/\epsilon^2)$ time, where x^* is the optimal solution and $\mathrm{nnz}(\mathbf{A})$ is the number of non-zero elements of matrix $\mathbf{A}$. To our knowledge, this is the first relative-error approximation algorithm for ridge regression that runs in $o(n^2p)$ time. In addition, we analyze the risk inflation bound of our algorithm and generalize our technique to the multiple response ridge regression problem. Finally, we show empirical results on both synthetic and real datasets.

4.1 Introduction

The learning algorithms of linear stochastic bandits employ ridge regression [93, 45, 1] as a key algorithmic component. Ridge regression is also widely applied in other areas of machine learning and statistics. It is an extension of the well-known ordinary linear regression by introducing an ℓ_2 regularization. Ridge regression can be formulated as the following optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 , \tag{4.1}$$

where $\mathbf{A}$ is an $n \times p$ sample-by-feature matrix called *design matrix*, $\mathbf{b}$ is an n-dimensional vector called *target vector*, and $\lambda \in \mathbb{R}^+$ is a regularization parameter.

A natural and popular approach [131] for solving Eq. (4.1) is to compute the unique minimizer $\mathbf{x}^*$ as follows

$$\mathbf{x}^* = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I}_n)^{-1}\mathbf{b}. \tag{4.2}$$

Computing $\mathbf{x}^*$ using Eq. (4.2) takes $O(n^2p + n^3)$ time, which is $O(n^2p)$ if $p \gg n$. If the data have high dimensional features, i.e., $p \gg n \gg 1$, this approach can be very slow.

In this chapter, we propose an algorithm for ridge regression that runs in $o(n^2p)$ and has provable relative-error approximation guarantees. More specifically, our algorithm computes a vector $\tilde{\mathbf{x}}$ satisfying that $\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2$ with high probability for full rank design matrix $\mathbf{A}$[1]. We show that the time complexity of our proposed algorithm is $O(\text{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2)$ where $\text{nnz}(\mathbf{A})$ is the number of non-zero entries of $\mathbf{A}$. Therefore, our algorithm is considerable faster than the $O(n^2p+n^3)$ time solver which uses Eq. (4.2) to compute $\mathbf{x}^*$, for the $p \gg n$ cases.

[1] Our algorithm can be extended to the case of general $\mathbf{A}$, see [42] for details

In addition, we prove a risk inflation bound of our algorithm under standard statistical assumptions. Further, we extend our algorithm to multiple ridge regression problem, where there are multiple target vectors. We also prove a relative-error bound on the output for this extension. We conduct experiments using both synthetic and real datasets. The experimental results agree with our theory and show that the proposed algorithm considerably improves the computational efficiency while produces accurate solutions.

4.1.1 Notations and Preliminaries

In this part, we describe the notations and definitions that are used later in this chapter. Let $\mathbf{I}_n$ denote the $n \times n$ identity matrix and let $\mathbf{0}_n$ denote the $n \times n$ zero matrix. Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times p}$ of rank r, let $\mathbf{M}_{(i)}$, where $i \in [n]$, denote the i-th row of $\mathbf{M}$ as a row vector. Let $\mathrm{nnz}(\mathbf{M})$ denote the number of non-zero entries of $\mathbf{M}$. Let $\|\mathbf{M}\|_F$ denote the Frobenius norm of $\mathbf{M}$ and let $\|\mathbf{M}\|_2$ denote the spectral norm of $\mathbf{M}$. Let $\sigma_i(\mathbf{M})$ denote the i-th largest singular value of $\mathbf{M}$ and let $\sigma_{\max}(\mathbf{M})$ and $\sigma_{\min}(\mathbf{M})$ denote the largest and smallest singular values of $\mathbf{M}$. The thin SVD of $\mathbf{M}$ is $\mathbf{M} = \mathbf{U}_M \mathbf{\Sigma}_M \mathbf{V}_M^T$, where $\mathbf{U}_M \in \mathbb{R}^{n \times r}$, $\mathbf{\Sigma}_M \in \mathbb{R}^{r \times r}$ and $\mathbf{V}_M \in \mathbb{R}^{p \times r}$. The Moore-Penrose pseudoinverse of $\mathbf{M}$ is a $p \times n$ matrix defined by $\mathbf{M}^\dagger = \mathbf{V}_M \mathbf{\Sigma}_M^{-1} \mathbf{U}_M^T$.

4.1.2 Oblivious subspace embedding

Our algorithm is based on the oblivious subspace embedding (OSE) technique. In the following, we review the definition of oblivious subspace embedding (OSE) and introduce the OSEs used in our algorithm.

Definition 4.1. *Given any $r > 0$, $\delta \in (0, 1)$ and $\epsilon \in (0, 1)$, we call a $t \times p$ random matrix $\mathbf{S}$ an (r, δ, ϵ)-OSE, if, for any rank*

r matrix $\mathbf{M} \in \mathbb{R}^{p \times m}$, the following holds simultaneously for all $\mathbf{z} \in \mathbb{R}^m$,

$$(1 - \epsilon) \left\| \mathbf{Mz} \right\|_2 \leq \left\| \mathbf{SMz} \right\|_2 \leq (1 + \epsilon) \left\| \mathbf{Mz} \right\|_2 ,$$

with probability at least $1 - \delta$.

Many types of random matrices have been shown to be OSEs including Gaussian matrices [49] and random sign matrices [129]. However, these types of matrices are dense matrices. It is clear that computing sketched matrix $\mathbf{SM}$ given $\mathbf{M} \in \mathbb{R}^{p \times m}$ takes $O(t \cdot \text{nnz}(\mathbf{M}))$ time for a dense OSE $\mathbf{S} \in \mathbb{R}^{t \times p}$. Hence, several work focused on finding $\mathbf{S}$ that allows fast matrix-vector multiplication in order to speed up the computation of the sketched matrix $\mathbf{SM}$ [46, 149, 109]. We refer readers to [23] for the development of this line of research.

Our algorithm uses a combination of two types of fast OSEs: *sparse embedding* and *subsampled randomized Hadamard transformation* (SRHT). The detailed definitions and properties sparse embedding and SRHT can be found in [42].

We consider the product $\mathbf{S} = \mathbf{\Phi}_{\text{srht}} \mathbf{\Phi}_{\text{sparse}}$, where $\mathbf{\Phi}_{\text{srht}}$ is a $t \times t'$ SRHT matrix and $\mathbf{\Phi}_{\text{sparse}}$ is a $t' \times p$ sparse embedding matrix. We show that if $t = O([\sqrt{r} + \sqrt{\log(p)}]^2 \log(r)/\epsilon)$ and $t' = O(r^2/\epsilon^2)$, the product $\mathbf{S}$ is an OSE.

Theorem 4.1. *Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $r > 0$, select integers $t' \geq 2\delta^{-1}(r^2 + r)/(2\epsilon/3 - \epsilon^2/9)^2$ and $t \geq 18\epsilon^{-1}[\sqrt{r} + \sqrt{8 \log(6p/\delta)}]^2 \log(6r/\delta)$. Let $\mathbf{\Phi}_{sparse}$ be a $t' \times p$ sparse embedding matrix and let $\mathbf{\Phi}_{srht}$ be a $t \times t'$ SRHT matrix. Then the product $\mathbf{S} = \mathbf{\Phi}_{srht} \mathbf{\Phi}_{sparse}$ is an (r, δ, ϵ)-OSE.*

Therefore, we see that, when $r \geq O(\log(p))$, the product OSE $\mathbf{S}$ has $t = O(r \log(r)/\epsilon)$ rows. This is smaller than using sparse embedding matrix alone. Finally, computing a sketched matrix $\mathbf{SM}$ given $\mathbf{M} \in \mathbb{R}^{p \times m}$ takes $O(\text{nnz}(\mathbf{M}) + mr^2 \log(r)/\epsilon^2)$ time. The proof of Theorem 4.1 is deferred to Section 4.7.

4.2 Algorithm and Main Results

In this section, we present our approximation algorithm for ridge regression (Algorithm 7). Then, we state our main result on the approximation guarantee of our algorithm (Theorem 4.2). In the rest of the chapter, we shall assume that the design matrix $\mathbf{A}$ is a full rank matrix. For the results on general $\mathbf{A}$, we refer readers to [42].

Algorithm. Algorithm 7 takes four inputs (1) design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, (2) target vector $\mathbf{b} \in \mathbb{R}^n$, (3) regularization parameter $\lambda > 0$ and (4) integer parameters t' and t. First, Algorithm 7 computes the sketched matrix $\mathbf{AS}^T$, where $\mathbf{S} \in \mathbb{R}^{t \times p}$ is defined to be the product of sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t' \times p}$ and SRHT matrix $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t \times t'}$, i.e. $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. Then, the algorithm computes the approximate solution $\tilde{\mathbf{x}}$ of ridge regression Eq. (4.1), by using the following key estimation of $\tilde{\mathbf{x}}$,

$$\tilde{\mathbf{x}} = \mathbf{A}^T(\mathbf{AS}^T)^{\dagger^T}(\lambda(\mathbf{AS}^T)^{\dagger^T} + \mathbf{AS}^T)^{\dagger}\mathbf{b}. \tag{4.3}$$

The procedure of computing $\mathbf{AS}^T$ and $\tilde{\mathbf{x}}$ is summarized in Algorithm 7.

Main result. We show that, with high probability, the output $\tilde{\mathbf{x}}$ of Algorithm 7 is a relative-error approximation to the optimal solution $\mathbf{x}^*$ of ridge regression.

Theorem 4.2. *Suppose that we are given a full rank design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, a target vector $\mathbf{b} \in \mathbb{R}^n$, a regularization parameter $\lambda > 0$, accuracy parameters $\epsilon \in (0,1)$ and $\delta \in (0,1)$. Assume that $n < p$. Select integers t', t such that $t' \geq 2\delta^{-1}(n^2 + n)/(\epsilon/6 - \epsilon^2/144)^2$ and $t \geq 72\epsilon^{-1}[\sqrt{n} + \sqrt{8\log(6p/\delta)}]^2 \log(6n/\delta)$. Run Algorithm 7 with inputs $\mathbf{A}$, $\mathbf{b}$, λ, t', t and let $\tilde{\mathbf{x}}$ denote the output of the algorithm. Then, with probability at least $1 - \delta$, we have*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2, \tag{4.4}$$

where $\mathbf{x}^*$ is the optimal solution of ridge regression in Eq. (4.1).

Algorithm 7 Fast relative-error approximation algorithm of ridge regression

```
Require: design matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ ($n$ samples with $p$ features), target vector
    $\mathbf{b} \in \mathbb{R}^n$, regularization parameter $\lambda > 0$, integer parameters $t'$ and $t$.
Ensure: approximate solution $\tilde{\mathbf{x}} \in \mathbb{R}^p$ to ridge regression problem Eq. (4.1).
Construct sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t' \times p}$.
Construct SRHT matrix $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t \times t'}$.
for each row $\mathbf{A}_{(i)}$ of $\mathbf{A}$ in arbitrary order do
    Compute $(\mathbf{AS}^T)_{(i)} \leftarrow (\mathbf{A}_{(i)} \mathbf{\Phi}^T_{\text{sparse}}) \mathbf{\Phi}^T_{\text{srht}}$
end for
Construct $\mathbf{AS}^T$ by concatenating row vectors $\{(\mathbf{AS}^T)_{(i)}\}_{i \in [n]}$.
Compute the pseudoinverse $(\mathbf{AS}^T)^\dagger$
Set $\tilde{\mathbf{x}} \leftarrow \mathbf{A}^T (\mathbf{AS}^T)^{\dagger^T} (\lambda (\mathbf{AS}^T)^{\dagger^T} + \mathbf{AS}^T)^\dagger \mathbf{b}$
return $\tilde{\mathbf{x}}$
```

In addition, if $t' = O(n^2/\epsilon^2)$ and $t = O(n \log(n)/\epsilon)$, the time complexity of Algorithm 7 is

$$O\left(\text{nnz}(\mathbf{A}) + n^3 \log\left(\frac{n}{\epsilon}\right) / \epsilon^2\right).$$

Running times. Define $t' = O(n^2/\epsilon^2)$ and $t = O(n \log(n)/\epsilon)$ according to Theorem 4.2. The time complexity of constructing sparse embedding matrix $\mathbf{\Phi}_{\text{sparse}}$ and SRHT matrix $\mathbf{\Phi}_{\text{srht}}$ is $O(p)$. Computing $\mathbf{AS}^T$ takes $O(\text{nnz}(\mathbf{A}) + nt' \log(t')) = O(\text{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2)$ time. The pseudoinverse of $\mathbf{AS}^T$ and $\lambda(\mathbf{AS}^T)^{\dagger^T} + \mathbf{AS}^T$ can be computed in $O(n^2 t) = O(n^3 \log(n)/\epsilon)$ time. Computing the product $(\mathbf{AS}^T)^{\dagger^T}(\lambda(\mathbf{AS}^T)^{\dagger^T} + \mathbf{AS}^T)^\dagger \mathbf{b}$ also takes $O(n^2 t) = O(n^3 \log(n)/\epsilon)$ time. Finally, multiplying $\mathbf{A}^T$ uses $O(\text{nnz}(\mathbf{A}))$ time. Summing up, the total running time is the sum of all these operations, which is $O(\text{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2)$.

Remarks. We note that the estimation method of $\tilde{\mathbf{x}}$ as in Eq. (4.3) holds for general OSEs $\mathbf{S}$, not necessarily limiting to the one used in Algorithm 7, i.e. $\mathbf{S} = \mathbf{\Phi}_{\text{srht}} \mathbf{\Phi}_{\text{sparse}}$. This fact is formalized in the following lemma.

Lemma 4.1. *Given a full rank matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ $(n < p)$, $\mathbf{b} \in \mathbb{R}^n$ and $\lambda > 0$. Suppose that $\mathbf{S} \in \mathbb{R}^{t \times p}$ is an $(n, \delta, \epsilon/4)$-OSE* *for* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$. *Then, with probability at least* $1-\delta$, *the approximation solution* $\tilde{\mathbf{x}}$ *obtained by Eq.* (4.3) *satisfies*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2 \, ,$$

where $\mathbf{x}^*$ *is the optimal solution to ridge regression Eq.* (4.1).

4.3 Risk Inflation Bound

In this section, we study the risk inflation of the approximate solution $\tilde{\mathbf{x}}$ returned by Algorithm 7 with respect to the optimal solution $\mathbf{x}^*$ of ridge regression. We begin with review the definition of risk of ridge regression. To properly define the risk, we need to that $\mathbf{A}$ and $\mathbf{b}$ have the linear relationship as follows

$$\mathbf{b} = \mathbf{A}\mathbf{x}_0 + \mathbf{e}, \tag{4.5}$$

where $\mathbf{x}_0 \in \mathbb{R}^p$ is an unknown vector which is assumed to be the "true" parameter and $\mathbf{e} \in \mathbb{R}^n$ is independent noise in each coordinate, with $\mathbf{E}[e_i] = 0$ and $\mathrm{Var}[e_i] = \sigma^2$. Under this assumption, the risk of any vector $\hat{\mathbf{b}} \in \mathbb{R}^n$ is given by

$$\mathrm{risk}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\mathbf{E}\left[\left\|\hat{\mathbf{b}} - \mathbf{A}\mathbf{x}_0\right\|_2^2\right],$$

where the expectation is taken over the randomness of noise [13].

The following theorem shows that, compared with the optimal solution $\mathbf{x}^*$, the approximate solution $\tilde{\mathbf{x}}$ returned by Algorithm 7 increases the risk by a small additive factor.

Theorem 4.3. *Suppose that* $n < p$. *Given a full rank design matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{b} \in \mathbb{R}^n$, $\lambda > 0$, $\epsilon \in (0,1)$ *and* $\lambda \in (0,1)$. *Assume that* $\mathbf{A}$ *and* $\mathbf{b}$ *have the linear relationship as in Eq.* (4.5). *Let* $\tilde{\mathbf{x}}$ *denote the output of Algorithm 7 with inputs* $\mathbf{A}$, $\mathbf{b}$, λ, $t' = \lceil 2\delta^{-1}(n^2+n)/(\epsilon/6 - \epsilon^2/144)^2 \rceil$ *and* $t = \left\lceil 72\epsilon^{-1}[\sqrt{n} + \sqrt{8\log(6p/\delta)}]^2 \log(6n/\delta) \right\rceil$.

Let $\mathbf{x}^*$ *denote the optimal solution of ridge regression. Then, with probability at least* $1-\delta$*,*

$$\mathrm{risk}(\tilde{\mathbf{b}}) \leq \mathrm{risk}(\mathbf{b}^*) + \frac{3\epsilon}{n} \|\mathbf{A}\|_2^2 \left(\|\mathbf{x}_0\|^2 + \sigma^2 \rho^2 \right), \tag{4.6}$$

where we define $\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{x}}$ *and* $\mathbf{b}^* = \mathbf{A}\mathbf{x}^*$*; we also define* $\rho^2 = \sum_{i\in[r]} \left(\frac{\sigma_i}{\sigma_i^2+\lambda}\right)^2$ *and* σ_i *is the* i*-th largest singular value of* $\mathbf{A}$*.*

4.4 Multiple Response Ridge Regression

In this section, we generalize our techniques to solve multiple response ridge regression [24].

The multiple response ridge regression problem is defined as follows. Given a design matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$, m target vectors (responses) $\mathbf{B} \in \mathbb{R}^{p\times m}$ and a regression parameter $\lambda > 0$, the multiple response regression problem is to find an $n \times m$ matrix $\mathbf{X}^*$ such that

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathbb{R}^{n\times m}}{\arg\min} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_F^2 + \lambda \|\mathbf{X}\|_F^2 . \tag{4.7}$$

The optimal solution of Eq. (4.7) is given by

$$\mathbf{X}^* = \mathbf{A}^T(\lambda \mathbf{I}_n + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}. \tag{4.8}$$

It is clear that Eq. (4.8) takes $O(n^2p+n^3+nm)$ time to compute, which is expensive if $p \gg n \gg 1$.

Next, we generalize our techniques to solve multiple response regression problem. The first step is to compute the sketched matrix $\mathbf{A}\mathbf{S}^T$, where $\mathbf{S} = \mathbf{\Phi}_{\mathrm{srht}}\mathbf{\Phi}_{\mathrm{sparse}}$. Notice that this step is identical to that of Algorithm 7, which uses one pass through $\mathbf{A}$. Then, we use the following generalized version of Eq. (4.3) to compute the approximate solution $\tilde{\mathbf{X}}$,

$$\tilde{\mathbf{X}} = \mathbf{A}^T(\mathbf{A}\mathbf{S}^T)^{\dagger T}(\lambda(\mathbf{A}\mathbf{S}^T)^{\dagger T} + \mathbf{A}\mathbf{S}^T)^{\dagger}\mathbf{B}, \tag{4.9}$$

which uses a second pass through $\mathbf{A}$. We show that the approximate solution $\tilde{\mathbf{X}}$ given by Eq. (4.9) is a relative-error approximation of $\mathbf{X}^*$ in the following theorem.

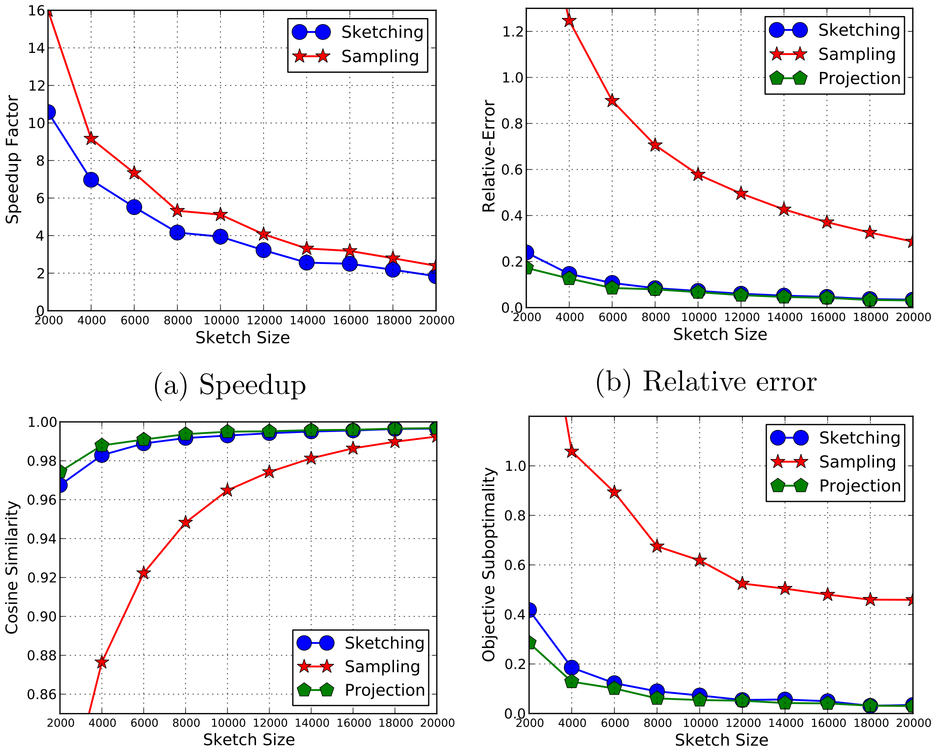
Theorem 4.4. *Given a full rank matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\lambda > 0$, *parameter* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$. *Select integers* t', t *such that* $t' \geq 2\delta^{-1}(n^2+n)/(\epsilon/6 - \epsilon^2/144)^2$ *and* $t \geq 72\epsilon^{-1}[\sqrt{n} + \sqrt{8\log(6p/\delta)}]^2 \log(6n/\delta)$. *Let* $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$, *where* $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t'\times p}$ *is a sparse embedding matrix and* $\mathbf{\Phi}_{srht} \in \mathbb{R}^{t\times t'}$ *is an SRHT matrix. Then, with probability at least* $1-\delta$, *we have*

$$\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F \leq \epsilon \left\|\mathbf{X}^*\right\|_F,$$

where $\tilde{\mathbf{X}}$ *is given by Eq.* (4.9) *and* $\mathbf{X}^*$ *is the optimal solution to multiple response ridge regression Eq.* (4.7). *In addition, the total time complexity of computing* $\mathbf{A}\mathbf{S}^T$ *and* $\tilde{\mathbf{X}}$ *is* $O(\mathrm{nnz}(\mathbf{A}) + n^3\log(\frac{n}{\epsilon})/\epsilon^2 + nm)$.

4.5 Experiments

Baselines. We compare the performance of our algorithm SKETCHING (Algorithm 7) to three baselines. The first baseline is the STANDARD algorithm, which computes the optimal solution using the dual space approach in Eq. (4.2). The other two baselines use popular randomized dimensionality reduction methods, including sampling and random projection. The SAMPLING algorithm simply samples a subset of t columns of $\mathbf{A}$ uniformly at random. The PROJECTION algorithm post-multiplies $\mathbf{A}$ by a random sign matrix $\mathbf{\Phi}_{\text{sign}}^T \in \mathbb{R}^{p\times t}$, with each entry of $\mathbf{\Phi}_{\text{sign}}$ having value chosen from $\{\pm 1/\sqrt{t}\}$ uniformly at random. Then, sketched matrices with t columns obtained by SAMPLING and PROJECTION are plugged in Eq. (4.3) to compute an approximate solution $\tilde{\mathbf{x}}$ of ridge regression. Finally, notice that $\mathbf{\Phi}_{\text{sign}}$



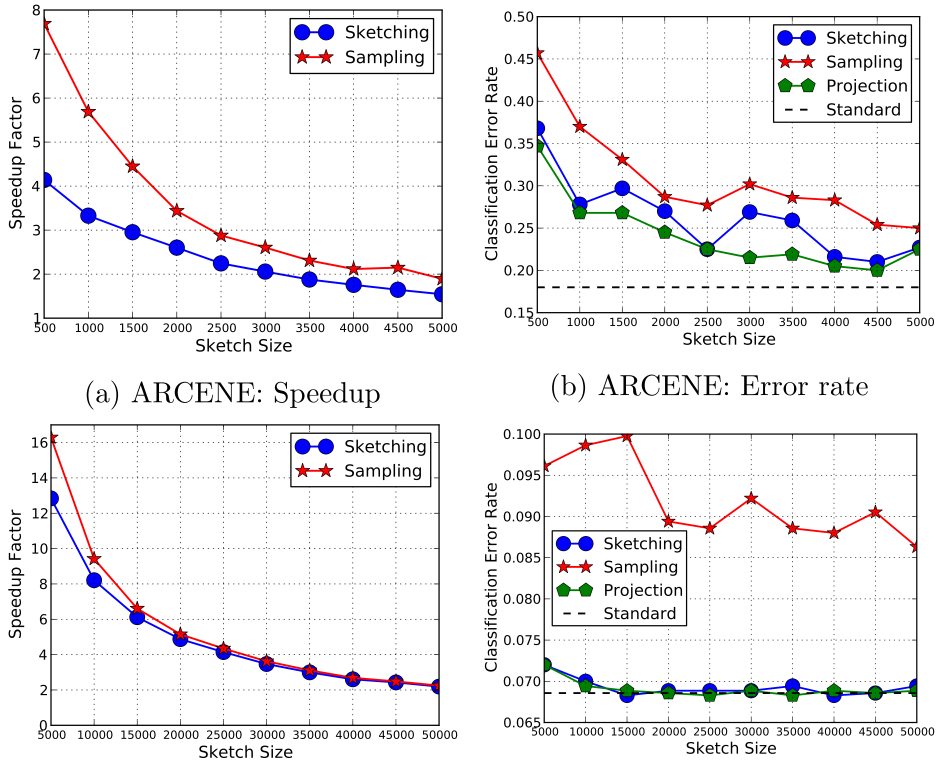
(a) Speedup

(b) Relative error

(c) Cosine similarity

(d) Objective suboptimality

Figure 4.1: Quality-of-approximation and running times on synthetic dataset



(a) ARCENE: Speedup

(b) ARCENE: Error rate

(c) DOROTHEA: Speed-up

(d) DOROTHEA: Error rate

Figure 4.2: Classification accuracy and running times on realworld datasets

is also an OSE for sufficiently large t [129] and therefore, by Lemma 4.1, the PROJECTION algorithm produces a relative-error approximation as well. However, for dense $\mathbf{A}$, computing $\mathbf{A}\mathbf{\Phi}_{\text{sign}}^{T}$ alone takes $O(tnp)$ time, which is even slower than the STANDARD algorithm when $t > n$. Hence, we do not compare its running time to other competing algorithms.

Implementation. Our implementation of Algorithm 7 is slightly different from its description in two places. First, Algorithm 7, and its analysis, uses the Walsh-Hadamard transformation (as a step of SRHT), while our implementation uses discrete Hartley transformation (DHT) [139]. DHT has a highly optimized implementation provided in FFTW package [59]. In addition, it is possible to show that DHT or other Fourier-type transformations have a guarantee similar to Walsh-Hadamard transformation [11, 149]. Second, we set $t' = 2t$, i.e. the sketch size of sparse embedding is two times larger than that of SRHT. Empirically, this setting offers a good trade-off between accuracy and speed. All competing algorithms are implemented using C++ and the experiments are conducted on a standard workstation using a single core.

4.5.1 Synthetic dataset

Setup. We generate the $n \times p$ design matrix $\mathbf{A}$ using the following method, such that each row (sample) of $\mathbf{A}$ contains an s-dimensional signal and p-dimensional noises. Specifically, we define $\mathbf{A} = \mathbf{M}\mathbf{\Sigma}\mathbf{V}^T + \alpha\mathbf{E}$. Here, $\mathbf{M}$ is an $n \times s$ matrix which represents the signals, and each entry $M_{ij} \sim \mathcal{N}(0,1)$ is an i.i.d Gaussian random variable. $\mathbf{\Sigma}$ is an $s \times s$ diagonal matrix and the diagonal entries are given by $\Sigma_{ii} = 1 - (i-1)/p$ for each $i \in [s]$. $\mathbf{V}$ is a $p \times n$ column orthonormal matrix which contains a random s-dimensional subspace of $\mathbb{R}^p$. Notice that $\mathbf{M}\mathbf{\Sigma}\mathbf{V}^T$ is a rank s matrix with linearly decreasing singular values. $\mathbf{E}$ is an $n \times p$ matrix which contributes the additive Gaussian noise $E_{ij} \sim \mathcal{N}(0,1)$. $\alpha > 0$ is a parameter chosen to balance the energy of signals $\mathbf{M}\mathbf{\Sigma}\mathbf{V}^T$ and the energy of noises $\mathbf{E}$. In this experiment, we choose $\alpha = 0.05$ which brings $\left\|\mathbf{M}\mathbf{\Sigma}\mathbf{V}^T\right\|_F \approx \alpha \left\|\mathbf{E}\right\|_F$. Then, we generate the target vector $\mathbf{x} \in \mathbb{R}^p$ with $x_i \sim \mathcal{N}(0,1)$. Finally, the target vector $\mathbf{b} \in \mathbb{R}^n$ is given by $\mathbf{b} = \mathbf{A}\mathbf{x} + \gamma\mathbf{e}$, where $e_i \sim \mathcal{N}(0,1)$ and $\gamma = 5$.

Metrics. We measure the performance of our algorithm and baselines both in terms of accuracy and speedup factor. More specifically, let $\mathbf{x}^*$ denote the optimal solution produced by the standard algorithm and let $\tilde{\mathbf{x}}$ denote the output vector returned by an approximation algorithm. To evaluate the accuracy of approximation, we compute three metrics: *relative error*: $\frac{\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2}$;

cosine similarity: $\frac{\tilde{\mathbf{x}}^T\mathbf{x}*}{\|\tilde{\mathbf{x}}\|_2\|\mathbf{x}^*\|_2}$; *objective suboptimality*: $\frac{f(\tilde{\mathbf{x}})}{f(\mathbf{x}^*)}-1$, with $f(\mathbf{x}) \triangleq \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$. In addition, the speedup factor is given by the ratio between the time used by STANDARD algorithm and that of a competing algorithm.

Results. In the experiment, we set $n = 500$, $p = 50000$ and $s = 50$. We run the competing algorithms with 10 different choices of t within range $[2000, 20000]$. The results are shown in Figure 4.1. Figure 4.1(a) reports the speed up of approximation algorithms with respect to the STANDARD algorithm. We see that our algorithm SKETCHING is slightly slower than the SAMPLING algorithm, but both of them speed up considerably with respect to the STANDARD algorithm. Figure 4.1(b), (c) and (d) plot the accuracy metrics of the competing algorithms. We see that indeed the accuracy of approximation improves as the sketch size t increases. In addition, both of our SKETCHING algorithm and the PROJECTION algorithm output a significantly more accurate solution than the SAMPLING algorithm. Notably, when the sketch size $t \approx 10000$, our algorithm SKETCHING has a relative-error smaller than 10%, cosine similarity larger than 99% and objective suboptimality less than 10%; meanwhile speeds up the computation about 4 times.

4.5.2 Realworld datasets

Setup. We also test the proposed algorithm on two binary classification datasets: ARCENE and DOROTHEA [69]. Both datasets are publicly available from the UCI repository [14]. ARCENE contains 200 samples (100 for training and 100 for testing) with 10000 real valued features. DOROTHEA consists of 1150 samples (800 for training and 350 for testing) with 100000 binary valued features. We apply ridge regression on both classification tasks by setting the responses to be +1 for positive examples and −1 for negative examples. We run ridge regression algorithms on the training data to compute the feature weights and measure the classification error rate on the testing data. For each dataset, we test 10 different choices of sketch size t and record the classification error rates and speed up factors of competing algorithms.

Results. The results[2] are shown in Figure 4.2. We observe that the classification error decreases as the sketch size t increases. It is also clear that, using the same sketch size t, SKETCHING and PROJECTION produce more accurate predictions than SAMPLING. On the other hand, SKETCHING and SAMPLING algorithms are considerably faster than the STANDARD algorithm. From the results, we see that our algorithm SKETCHING substantially speeds up the computation, while attains a very small increase in error rate. For ARCENE dataset, when $t \approx 3000$, SKETCHING accelerates the computation by 2.1 times while increases the error rate by 4.5%; and, for DOROTHEA dataset, when $t \approx 20000$, the speedup of SKETCHING is about 4.1 times and the error rate is almost the same to the STANDARD algorithm.

4.6 Conclusions

We presented an efficient relative-error approximation algorithm for ridge regression for $p \gg n$ cases. Our algorithm runs in $\tilde{O}(\text{nnz}(A) + n^3/\epsilon^2)$ time, which is substantially faster than the standard $O(n^2p+n^3)$ algorithm for large p instances. In addition, we analyzed the risk inflation of our algorithm and extended our techniques to design fast relative-error approximation algorithms for multiple response ridge regression and structured ridge regression. We reported experimental results of our algorithm on both synthetic and real datasets, which supported our analysis and demonstrated good practical performance.

[2] The results of relative-error are deferred to Section 4.11.

4.7 Proof of Theorem 4.1

For convenience, we restate the theorem in the following.

Theorem 4.1. *Given $\epsilon \in (0,1)$, $\delta \in (0,1)$ and $r > 0$, select integers $t' \geq 2\delta^{-1}(r^2+r)/(2\epsilon/3 - \epsilon^2/9)^2$ and $t \geq 18\epsilon^{-1}[\sqrt{r} + \sqrt{8\log(6p/\delta)}]^2 \log(6r/\delta)$. Let $\mathbf{\Phi}_{sparse}$ be a $t' \times p$ sparse embedding matrix and let $\mathbf{\Phi}_{srht}$ be a $t \times t'$ SRHT matrix. Then the product $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$ is an (r, δ, ϵ)-OSE.*

Proof. By [109, Theorem 3], $\mathbf{\Phi}_{\text{sparse}} \in \mathbb{R}^{t' \times p}$ is an $(r, \delta/2, \epsilon/3)$-OSE. Similarly, by [149, Lemma 4.1], $\mathbf{\Phi}_{\text{srht}} \in \mathbb{R}^{t' \times t}$ is also an $(r, \delta/2, \epsilon/3)$-OSE.

Now fix an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{p \times m}$ of rank r. Since $\mathbf{\Phi}_{\text{sparse}}$ is an $(r, \delta/2, \epsilon/3)$-OSE, by Definition 4.1, we have

$$(1 - \epsilon/3) \left\|\mathbf{Mz}\right\|_2 \leq \left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2 \leq (1 + \epsilon/3) \left\|\mathbf{Mz}\right\|_2, \quad (4.10)$$

holds for all $\mathbf{z} \in \mathbb{R}^m$ simultaneously with probability at least $1 - \delta/2$. Eq. (4.10) also implies that $\text{rank}(\mathbf{\Phi}_{\text{sparse}}\mathbf{M}) = \text{rank}(\mathbf{M}) = r$. Now, conditioning on the event that Eq. (4.10) holds, and using the fact that $\mathbf{\Phi}_{\text{srht}}$ is an $(r, \delta/2, \epsilon/3)$-OSE, we have

$$(1-\epsilon/3) \left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2 \leq \left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 \leq (1+\epsilon/3) \left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2, \quad (4.11)$$

holds for all $\mathbf{z} \in \mathbb{R}^m$ simultaneously with probability at least $1 - \delta/2$. When both Eq. (4.10) and Eq. (4.11) hold, we have, for any $\mathbf{z} \in \mathbb{R}^m$,

$$\begin{aligned} \left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 &\leq (1 + \epsilon/3) \left\|\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz}\right\|_2 \\ &\leq (1 + \epsilon/3)^2 \left\|\mathbf{Mz}\right\|_2 \\ &\leq (1 + \epsilon) \left\|\mathbf{Mz}\right\|_2, \end{aligned} \quad (4.12)$$

where we have used Eq. (4.10), Eq. (4.11) and the fact that $\epsilon < 1$. The other direction: $\left\|\mathbf{\Phi}_{\text{srht}}(\mathbf{\Phi}_{\text{sparse}}\mathbf{Mz})\right\|_2 \geq (1 - \epsilon) \left\|\mathbf{Mz}\right\|_2$ can be proved using the same method. Now, notice that, by union bound, the probability that both Eq. (4.10) and Eq. (4.11) hold simultaneously for any $\mathbf{z}$ is at least $1 - \delta$. This concludes our proof. □

4.8 Proof of Lemma 4.1 and Theorem 4.2

From this point on, we denote the thin SVD of design matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$ of full rank by $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, with $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{p\times n}$. We also denote the SVD of $\mathbf{SV}$ by $\mathbf{SV} = \mathbf{U}_\phi\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T$, with $\mathbf{U}_\phi \in \mathbb{R}^{t\times n}$, $\mathbf{\Sigma}_\phi \in \mathbb{R}^{n\times n}$ and $\mathbf{V}_\phi \in \mathbb{R}^{n\times n}$. Notice that $\mathbf{V}_\phi$ is an $n\times n$ unitary matrix and therefore $\mathbf{V}_\phi\mathbf{V}_\phi^T = \mathbf{I}_n$. We will frequently use the following property of the pseudoinverse of matrix product.

Fact 4.1. *For any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$, we have $(\mathbf{AB})^\dagger = \mathbf{B}^\dagger\mathbf{A}^\dagger$, if at least one of the following holds.*

1. *$\mathbf{A}$ has orthonormal columns.*
2. *$\mathbf{B}$ has orthonormal rows.*
3. *$\mathbf{A}$ has full column rank and $\mathbf{B}$ has full row rank.*

By Fact 4.1, we immediate obtain the following lemma.

Lemma 4.2. *Suppose that $\mathbf{SV}$ is full rank, then the pseudoinverse of $\mathbf{AS}^T$ is given by*

$$(\mathbf{AS}^T)^\dagger = (\mathbf{SV})^{\dagger T}\mathbf{\Sigma}^{-1}\mathbf{U}^T.$$

The first step of our proof is to represent $\mathbf{x}^*$ and $\tilde{\mathbf{x}}$ in a form that is easier to work with.

Lemma 4.3. *Let the optimal solution of ridge regression $\mathbf{x}^*$ be defined as in Eq. (4.2). We have*

$$\mathbf{x}^* = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b},$$

where $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$.

Proof. Notice that $\mathbf{A}$ is a full rank matrix. Therefore $\mathbf{U}$ is an orthonormal matrix. By definition of $\mathbf{x}^*$, we have

$$\begin{aligned}\mathbf{x}^* &= \mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \lambda\mathbf{I})^{-1}\mathbf{b}\\ &= \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T(\mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T + \lambda\mathbf{U}\mathbf{U}^T)^{-1}\mathbf{b}\\ &= \mathbf{V}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2 + \lambda\mathbf{I})^{-1}\mathbf{U}^T\mathbf{b}\\ &= \mathbf{V}(\boldsymbol{\Sigma} + \lambda\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{U}^T\mathbf{b}.\end{aligned}$$

□

Lemma 4.4. *Define matrix* $\tilde{\mathbf{G}} = \lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$*. Let* $\tilde{\mathbf{x}}$ *be defined as in Eq.* (4.3)*. Suppose that* $\mathbf{S}\mathbf{V}$ *is full rank. Then, we have that* $\tilde{\mathbf{G}}$ *is full rank and that*

$$\tilde{\mathbf{x}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{b}.$$

Proof. By the construction of $\tilde{\mathbf{x}}$, we have

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{A}^T\left((\mathbf{A}\mathbf{S}^T)^\dagger\right)^T\left(\lambda\left((\mathbf{A}\mathbf{S}^T)^\dagger\right)^T + \mathbf{A}\mathbf{S}^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger\left(\lambda\mathbf{U}\boldsymbol{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger + \mathbf{U}\boldsymbol{\Sigma}(\mathbf{S}\mathbf{V})^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}(\mathbf{S}\mathbf{V})^\dagger\left(\lambda\mathbf{U}\boldsymbol{\Sigma}^{-1}(\mathbf{S}\mathbf{V})^\dagger + \mathbf{U}\boldsymbol{\Sigma}(\mathbf{S}\mathbf{V})^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T\left(\lambda\mathbf{U}\boldsymbol{\Sigma}^{-1}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T + \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi\mathbf{U}_\phi^T\right)^\dagger\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1}\mathbf{U}_\phi^T\mathbf{U}_\phi\left(\lambda\boldsymbol{\Sigma}^{-1}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1} + \boldsymbol{\Sigma}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi\right)^\dagger\mathbf{U}^T\mathbf{b}\\ &= \mathbf{V}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1}\left(\lambda\boldsymbol{\Sigma}^{-1}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1} + \boldsymbol{\Sigma}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi\right)^\dagger\mathbf{U}^T\mathbf{b}, \qquad (4.13)\end{aligned}$$

where we have repeatedly used Fact 4.1 and Lemma 4.2.

Define $\mathbf{T}_1 = \lambda\boldsymbol{\Sigma}^{-1}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^{-1} + \boldsymbol{\Sigma}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi$. Next, we show that $\text{rank}(\mathbf{T}_1) = n$. To see this, we define $\mathbf{T}_2 = \lambda\mathbf{I} + \boldsymbol{\Sigma}\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^2\mathbf{V}_\phi^T\boldsymbol{\Sigma}$ and notice that $\mathbf{T}_2 = \mathbf{T}_1(\boldsymbol{\Sigma}_\phi\mathbf{V}_\phi^T\boldsymbol{\Sigma})$. Since $\lambda > 0$, it is clear that $\mathbf{T}_2$ is a positive definite matrix and therefore $\text{rank}(\mathbf{T}_2) = n$.

Now notice that $\text{rank}(\boldsymbol{\Sigma}_\phi\mathbf{V}_\phi^T\boldsymbol{\Sigma}) = \text{rank}(\boldsymbol{\Sigma}_\phi) = n$. Hence, we have

$$\text{rank}(\mathbf{T}_1) = \text{rank}(\mathbf{T}_1(\boldsymbol{\Sigma}_\phi\mathbf{V}_\phi^T\boldsymbol{\Sigma})) = \text{rank}(\mathbf{T}_2) = n.$$

Then, using Fact 4.1 on $\mathbf{\Sigma}_\phi^\dagger$ and $\mathbf{T}_1^\dagger$, we have

$$\begin{aligned}(4.13) &= \mathbf{V}\mathbf{V}_\phi \left(\lambda \mathbf{\Sigma}^{-1}\mathbf{V}_\phi + \mathbf{\Sigma}\mathbf{V}_\phi \mathbf{\Sigma}_\phi^2\right)^\dagger \mathbf{U}^T\mathbf{b} \\ &= \mathbf{V}\left(\lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}\mathbf{V}_\phi\mathbf{\Sigma}_\phi^2\mathbf{V}_\phi^T\right)^\dagger \mathbf{U}^T\mathbf{b} \\ &= \mathbf{V}\tilde{\mathbf{G}}^\dagger\mathbf{U}^T\mathbf{b},\end{aligned}$$

where we have used Fact 4.1 again and that $\mathbf{S}\mathbf{V} = \mathbf{U}_\phi\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T$. Finally, the rank of $\tilde{\mathbf{G}}$ is given by

$$\operatorname{rank}(\tilde{\mathbf{G}}) = \operatorname{rank}(\mathbf{T}_1\mathbf{\Sigma}_\phi\mathbf{V}_\phi^T) = \operatorname{rank}(\mathbf{T}_1) = n.$$

Hence the pseudoinverse of $\tilde{\mathbf{G}}$ equals to its inverse, i.e. $\tilde{\mathbf{G}}^\dagger = \tilde{\mathbf{G}}^{-1}$, and this concludes our proof of the lemma. □

From Lemma 4.3 and Lemma 4.4, we see that $\tilde{\mathbf{x}}$ admits a representation that is very similar to $\mathbf{x}^*$. It is clear that the key difference between $\tilde{\mathbf{x}}$ and $\mathbf{x}^*$ comes from that of $\tilde{\mathbf{G}}$ and $\mathbf{G}$.

The next lemma (Lemma 4.5) is our key technical lemma, which shows that $\tilde{\mathbf{G}}$ is closely related to $\mathbf{G}$ in the sense that $\mathbf{G}^{-1}$ is an approximate matrix inversion of $\tilde{\mathbf{G}}$.

Lemma 4.5. *Given $\epsilon \in (0, 1/4)$ and $\delta \in (0,1)$. Let $\mathbf{S}$ be an (n, δ, ϵ)-OSE. Let $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$ and $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$. Notice that $\mathbf{G}$ is invertible and define $\mathbf{R} = \mathbf{G}^{-1}\tilde{\mathbf{G}} - \mathbf{I}$. Then, with probability at least $1-\delta$, we have (a) $\mathbf{S}\mathbf{V}$ is a full rank matrix, (b) $\|\mathbf{R}\|_2 \le 2\epsilon + \epsilon^2$, and (c)*

$$\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \le \frac{2\epsilon+\epsilon^2}{1-(2\epsilon+\epsilon^2)}.$$

To prove Lemma 4.5, we need two lemmas from linear algerbra and the theory of OSEs. First, we use the following property on the stability of singular values, which can be regarded as a generalization of Weyl's inequality to singular values of non-Hermitian matrices.

Lemma 4.6. [144, Exercise 1.3.22 (iv)] *Let* $\mathbf{C} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{D} \in \mathbb{R}^{m\times n}$ *be two matrices of the same size. Then, for all* $i \in [\min\{m,n\}]$,

$$|\sigma_i(\mathbf{C}+\mathbf{D}) - \sigma_i(\mathbf{C})| \leq \|\mathbf{D}\|_2 .$$

Proof. Let $\mathbf{C}' = \begin{bmatrix} \mathbf{0}_m & \mathbf{C} \\ \mathbf{C}^T & \mathbf{0}_n \end{bmatrix}$, $\mathbf{D}' = \begin{bmatrix} \mathbf{0}_m & \mathbf{D} \\ \mathbf{D}^T & \mathbf{0}_n \end{bmatrix}$, then all of $\mathbf{C}'$, $\mathbf{D}'$ and $\mathbf{C}'+\mathbf{D}'$ are $(m+n)\times(m+n)$ Hermitian matrices and also

$$\sigma_i(\mathbf{C}) = \sigma_{2i}(\mathbf{C}'), \quad \sigma_i(\mathbf{C}+\mathbf{D}) = \sigma_{2i}(\mathbf{C}'+\mathbf{D}') \quad \forall i \in [\min\{m,n\}]; \tag{4.14}$$

$$\|\mathbf{D}\|_2 = \|\mathbf{D}'\|_2 . \tag{4.15}$$

Using Weyl's inequality [see 144, Section 1.3] on the singular values of Hermitian matrices, we have that

$$|\sigma_i(\mathbf{C}'+\mathbf{D}') - \sigma_i(\mathbf{C}')| \leq \|\mathbf{D}'\|_2 , \quad \forall i \in [m+n]. \tag{4.16}$$

Thus we have that:

$$|\sigma_i(\mathbf{C}+\mathbf{D}) - \sigma_i(\mathbf{C})| \leq \|\mathbf{D}\|_2 , \quad \forall i \in [\min\{m,n\}]. \tag{4.17}$$

□

The second ingredient we needed is the following characterization of OSEs.

Theorem 4.5. *Let* $\mathbf{V} \in \mathbb{R}^{p\times r}$ *be a column orthonormal matrix. Let* $\mathbf{S} \in \mathbb{R}^{t\times p}$ *be an* (r,δ,ϵ)*-OSE. Then, with probability* $1-\delta$ *over the choices of* $\mathbf{S}$*, we have that (a)* $\mathbf{SV}$ *is a full rank matrix and (b) for all* $i \in [r]$*, the* i*-th largest singular value of* $\mathbf{SV}$ *is bounded by*

$$|1 - \sigma_i(\mathbf{SV})| \leq \epsilon. \tag{4.18}$$

Proof. By the definition of OSE (Definition 4.1), with probability $1-\delta$, we have

$$(1-\epsilon)\left\|\mathbf{V}\mathbf{z}\right\|_2 \leq \left\|\mathbf{S}\mathbf{V}\mathbf{z}\right\|_2 \leq (1+\epsilon)\left\|\mathbf{V}\mathbf{z}\right\|_2 \quad \text{for all } \mathbf{z} \in \mathbb{R}^p. \tag{4.19}$$

In the rest of the proof, we assume that Eq. (4.19) holds.

First, we show that $\mathbf{SV}$ is a rank r matrix. Since $\mathbf{V}$ is a rank r matrix, it suffices to show that $\ker(\mathbf{SV}) = \ker(\mathbf{V})$. It is easy to see that $\ker(\mathbf{V}) \subseteq \ker(\mathbf{SV})$: Indeed, for any $\mathbf{z} \in \ker(\mathbf{V})$, we have $\mathbf{SVz} = \mathbf{S}(\mathbf{Vz}) = \mathbf{0}$, which means that $\mathbf{z} \in \ker(\mathbf{SV})$.

Now, we prove the other direction: $\ker(\mathbf{SV}) \subseteq \ker(\mathbf{V})$. Consider any vector $\mathbf{z} \in \ker(\mathbf{SV})$. Using Eq. (4.19) and the fact that $\mathbf{SVz} = \mathbf{0}$, we have

$$0 = \left\|\mathbf{SVz}\right\|_2 \geq (1-\epsilon)\left\|\mathbf{Vz}\right\|_2 \geq 0.$$

Therefore, the above equality holds which means that $\mathbf{Vz} = \mathbf{0}$ and $\mathbf{z} \in \ker(\mathbf{V})$.

Next, we prove part (b) of the theorem. Fix an arbitrary $i \in [r]$. Let $\mathbf{v}_i^{\phi}$ denote the i-th right singular vector of $\mathbf{SV}$. Now, using Eq. (4.19) and the definition of singular value, we have

$$\begin{aligned}
\sigma_i(\mathbf{SV}) &= \left\|\mathbf{SV}\mathbf{v}_i^{\phi}\right\|_2 && (4.20)\\
&\leq (1+\epsilon)\left\|\mathbf{V}\mathbf{v}_i^{\phi}\right\|_2 && (4.21)\\
&= (1+\epsilon)\left\|\mathbf{v}_i^{\phi}\right\|_2 && (4.22)\\
&= (1+\epsilon), && (4.23)
\end{aligned}$$

where Eq. (4.20) follows from the definition of singular value, Eq. (4.21) follows form Eq. (4.19) and Eq. (4.22) is obtained by dropping $\mathbf{V}$ which does not change l_2 norm. The other direction: $\sigma_i(\mathbf{SV}) \geq (1-\epsilon)$ can be proved using the same method by using the other side of Eq. (4.19).

Hence, we have proved that $\sigma_i(\mathbf{SV}) \in [1-\epsilon, 1+\epsilon]$ for all $i \in [r]$, which concludes the part (b) of the theorem. □

Proof of Lemma 4.5. Since $\mathbf{S}$ is an (r, δ, ϵ)-OSE. By Theorem 4.5, we have that, with probability $1 - \delta$, $\mathbf{SV}$ is a full rank matrix and all singular values of $\mathbf{SV}$ are bounded in $[1-\epsilon, 1+\epsilon]$. In the rest of the proof, we assume this holds. And this already proves part (a) of the lemma.

We start with bounding $\|\mathbf{R}\|_2$. By the definition of $\mathbf{R}$, we have

$$\begin{aligned}
\|\mathbf{R}\|_2 &= \left\|\mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})\right\|_2 \\
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}((\mathbf{SV})^T(\mathbf{SV}) - \mathbf{I})\right\|_2 \\
&\leq \left\|(\lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^2\mathbf{V}_\phi^T - \mathbf{I}\right\|_2 \\
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\mathbf{V}_\phi\boldsymbol{\Sigma}_\phi^2\mathbf{V}_\phi^T - \mathbf{V}_\phi\mathbf{V}_\phi^T\right\|_2 \\
&= \left\|(\lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\right\|_2 \left\|\boldsymbol{\Sigma}_\phi^2 - \mathbf{I}\right\|_2 \\
&\leq \max_i \frac{\sigma_i}{\lambda\sigma_i^{-1} + \sigma_i}((1+\epsilon)^2 - 1) \\
&\leq 2\epsilon + \epsilon^2, \qquad (4.24)
\end{aligned}$$

where we have used the fact that $\mathbf{V}_\phi$ is a unitary matrix and dropped terms that do not change spectral norm.

Now, we apply Lemma 4.6 by setting $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{R}$. Then, for all $i \in [r]$, we have

$$\sigma_i(\mathbf{I} + \mathbf{R}) \geq 1 - \|\mathbf{R}\|_2 \geq 1 - (2\epsilon + \epsilon^2). \qquad (4.25)$$

Hence, we have

$$\begin{aligned}
\left\|(\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\right\|_2 &\leq \left\|(\mathbf{I} + \mathbf{R})^{-1}\right\|_2 \|\mathbf{R}\|_2 \\
&\leq (\sigma_{\min}(\mathbf{I} + \mathbf{R}))^{-1} \|\mathbf{R}\|_2 \\
&\leq \frac{2\epsilon + \epsilon^2}{1 - (2\epsilon + \epsilon^2)}.
\end{aligned}$$

□

We are now ready to prove our main results: Lemma 4.1 and Theorem 4.2. For reader's convenience, we first restate them in the following.

Lemma 4.1. *Given a full rank matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$ *($n < p$),* $\mathbf{b} \in \mathbb{R}^n$ *and* $\lambda > 0$*. Suppose that* $\mathbf{S} \in \mathbb{R}^{t\times p}$ *is an* $(n, \delta, \epsilon/4)$*-OSE for* $\epsilon \in (0, 1)$ *and* $\delta \in (0, 1)$*. Then, with probability at least* $1-\delta$*, the approximation solution* $\tilde{\mathbf{x}}$ *obtained by Eq.* (4.3) *satisfies*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \le \epsilon \|\mathbf{x}^*\|_2 ,$$

where $\mathbf{x}^*$ *is the optimal solution to ridge regression Eq.* (4.1)*.*

Proof. Let $\epsilon' = \epsilon/4$ and recall the definition $\mathbf{R} = \mathbf{G}^{-1}\tilde{\mathbf{G}} - \mathbf{I}$. Since $\mathbf{S}$ is an (n, δ, ϵ')-OSE. By Lemma 4.5, with probability $1-\delta$, we have that $\mathbf{SV}$ is a full rank matrix and that $\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \le \frac{2\epsilon'+\epsilon'^2}{1-(2\epsilon'+\epsilon'^2)}$. In the rest of the proof, we assume that this event happens.

Since $\mathbf{SV}$ is a full rank matrix. Applying Lemma 4.3 and Lemma 4.4, we have

$$\begin{aligned} \|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 &= \left\|\mathbf{V}(\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{b}\right\|_2 \\ &= \left\|(\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{b}\right\|_2 . \qquad (4.26) \end{aligned}$$

where we have dropped the unitary term $\mathbf{V}$ which does not change l_2 norm.

Next, we write $\tilde{\mathbf{G}} = \mathbf{G}(\mathbf{I}+\mathbf{R})$. This means that $\tilde{\mathbf{G}}^{-1} = (\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}$. Therefore,

$$\begin{aligned} (4.26) &= \left\|\left((\mathbf{I}+\mathbf{R})^{-1} - \mathbf{I}\right)\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \\ &= \left\|-(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \qquad (4.27) \\ &\le \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}\right\|_2 \\ &= \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \|\mathbf{x}^*\|_2 \\ &\le \frac{2\epsilon'+\epsilon'^2}{1-(2\epsilon'+\epsilon'^2)} \|\mathbf{x}^*\|_2 \qquad (4.28) \\ &\le 4\epsilon' \|\mathbf{x}^*\|_2 = \epsilon \|\mathbf{x}^*\|_2 . \end{aligned}$$

where Eq. (4.27) follows from matrix inversion lemma, i.e. $\mathbf{C}^{-1} - \mathbf{D}^{-1} = -\mathbf{C}^{-1}(\mathbf{C}-\mathbf{D})\mathbf{D}^{-1}$ for any squared matrices $\mathbf{C}$ and $\mathbf{D}$

of the same size, and Eq. (4.28) follows from the assumption on $\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2$. □

Theorem 4.2. *Suppose that we are given a full rank design matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$*, a target vector* $\mathbf{b} \in \mathbb{R}^n$*, a regularization parameter* $\lambda > 0$*, accuracy parameters* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$*. Assume that* $n < p$*. Select integers* t', t *such that* $t' \geq 2\delta^{-1}(n^2+n)/(\epsilon/6-\epsilon^2/144)^2$ *and* $t \geq 72\epsilon^{-1}[\sqrt{n}+\sqrt{8\log(6p/\delta)}]^2\log(6n/\delta)$*. Run Algorithm 7 with inputs* $\mathbf{A}$*,* $\mathbf{b}$*,* λ*,* t'*,* t *and let* $\tilde{\mathbf{x}}$ *denote the output of the algorithm. Then, with probability at least* $1-\delta$*, we have*

$$\|\tilde{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \epsilon \|\mathbf{x}^*\|_2 , \tag{4.4}$$

where $\mathbf{x}^*$ *is the optimal solution of ridge regression in Eq.* (4.1)*.*

In addition, if $t' = O(n^2/\epsilon^2)$ *and* $t = O(n\log(n)/\epsilon)$*, the time complexity of Algorithm 7 is*

$$O\left(\mathrm{nnz}(\mathbf{A}) + n^3\log\left(\frac{n}{\epsilon}\right)/\epsilon^2\right).$$

Proof. It is easy to see that the solution $\tilde{\mathbf{x}}$ returned by Algorithm 7 is given by Eq. (4.3) with $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$. Therefore, the bound on $\|\tilde{\mathbf{x}} - \mathbf{x}\|_2$ follows immediately from Lemma 4.1 and Theorem 4.1 which shows that $\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$ is an $(r, \delta, \epsilon/4)$-OSE. And the running time analysis of Algorithm 7 is given in Section 4.2. □

4.9 Proof of Theorem 4.3

Theorem 4.3. *Suppose that* $n < p$*. Given a full rank design matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$*,* $\mathbf{b} \in \mathbb{R}^n$*,* $\lambda > 0$*,* $\epsilon \in (0,1)$ *and* $\lambda \in (0,1)$*. Assume that* $\mathbf{A}$ *and* $\mathbf{b}$ *have the linear relationship as in Eq.* (4.5)*. Let* $\tilde{\mathbf{x}}$ *denote the output of Algorithm 7 with inputs* $\mathbf{A}$*,* $\mathbf{b}$*,* λ*,* $t' = \lceil 2\delta^{-1}(n^2+n)/(\epsilon/6-\epsilon^2/144)^2\rceil$ *and* $t = \left\lceil 72\epsilon^{-1}[\sqrt{n}+\sqrt{8\log(6p/\delta)}]^2\log(6n/\delta)\right\rceil$*. Let* $\mathbf{x}^*$ *denote the optimal solution of ridge regression. Then, with* *probability at least* $1-\delta$,

$$\mathrm{risk}(\tilde{\mathbf{b}}) \leq \mathrm{risk}(\mathbf{b}^*) + \frac{3\epsilon}{n}\left\|\mathbf{A}\right\|_2^2\left(\left\|\mathbf{x}_0\right\|^2 + \sigma^2\rho^2\right), \tag{4.6}$$

where we define $\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{x}}$ *and* $\mathbf{b}^* = \mathbf{A}\mathbf{x}^*$*; we also define* $\rho^2 = \sum_{i\in[r]}\left(\frac{\sigma_i}{\sigma_i^2+\lambda}\right)^2$ *and* σ_i *is the i-th largest singular value of* $\mathbf{A}$*.*

Proof. Let $\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{x}}$ denote the prediction using approximated $\tilde{\mathbf{x}}$ returned by Algorithm 7. Let $\mathbf{b}^* = \mathbf{A}\mathbf{x}^*$ denote the prediction using optimal solution $\mathbf{x}^*$ of Eq. (4.1). Define $\mathbf{b}_0 = \mathbf{A}\mathbf{x}_0$. Let the thin SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$.

Using the classical bias-variance decomposition [13], for any $\hat{\mathbf{b}} \in \mathbb{R}^n$, we have

$$\mathrm{risk}(\hat{\mathbf{b}}) = \mathrm{bias}(\hat{\mathbf{b}}) + \mathrm{var}(\hat{\mathbf{b}}), \tag{4.29}$$

where we define

$$\mathrm{bias}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\left\|\mathbf{E}\left[\hat{\mathbf{b}}\right] - \mathbf{b}_0\right\|_2^2 \quad \text{and} \quad \mathrm{var}(\hat{\mathbf{b}}) \triangleq \frac{1}{n}\mathbf{E}\left[\left\|\hat{\mathbf{b}} - \mathbf{E}\left[\hat{\mathbf{b}}\right]\right\|_2^2\right]$$

as the bias component and the variance component, respectively.

By Lemma 4.3, we have $\mathbf{x}^* = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{b}$, where $\mathbf{G} = \lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}$. And by Lemma 4.4, we have $\tilde{\mathbf{x}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{b}$, where $\tilde{\mathbf{G}} = \lambda\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$. Also notice that $\left\|\mathbf{G}^{-1}\right\|_F^2 = \rho^2$. Using these definition, we first bound the variance component $\mathrm{var}(\tilde{\mathbf{b}})$. We have

$$\begin{aligned}
\mathrm{var}(\tilde{\mathbf{b}}) &= \frac{1}{n}\mathbf{E}\left[\left\|\tilde{\mathbf{b}} - \mathbf{E}\left[\hat{\mathbf{b}}\right]\right\|_2^2\right] \\
&= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{E}\left[\mathbf{A}\tilde{\mathbf{x}}\right]\right\|_2^2\right] \\
&= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{A}\mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right\|_2^2\right] \\
&= \frac{1}{n}\mathbf{E}\left[\left\|\mathbf{U}\boldsymbol{\Sigma}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right\|_2^2\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}\mathbf{E}\left[\operatorname{tr}\left(\mathbf{e}^T\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\mathbf{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\right)\right]\\
&= \frac{1}{n}\mathbf{E}\left[\operatorname{tr}\left(\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\mathbf{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{e}\mathbf{e}^T\right)\right]\\
&= \frac{\sigma^2}{n}\operatorname{tr}\left(\mathbf{U}(\tilde{\mathbf{G}}^{-1})^T\mathbf{\Sigma}^2\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\right)\\
&= \frac{\sigma^2}{n}\operatorname{tr}\left((\tilde{\mathbf{G}}^{-1})^T\mathbf{\Sigma}^2\tilde{\mathbf{G}}^{-1}\right)\\
&= \frac{\sigma^2}{n}\left\|\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F^2,
\end{aligned}
$$

where we have repeatedly used the cyclic property of trace of matrix product. Similarly, one can show that

$$\operatorname{var}(\mathbf{b}^*) = \frac{\sigma^2}{n}\left\|\mathbf{\Sigma}\mathbf{G}^{-1}\right\|_F^2.$$

Now we recall the definition $\mathbf{R} = \mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})$ and write $\tilde{\mathbf{G}}^{-1} = (\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}$. Then, we have

$$
\begin{aligned}
\left\|\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F &= \left\|\mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1}\right\|_F\\
&= \left\|\mathbf{\Sigma}\mathbf{G}^{-1} - \mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\right\|_F \qquad (4.30)\\
&\le \left\|\mathbf{\Sigma}\mathbf{G}^{-1}\right\|_F + \left\|\mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\right\|_F \qquad (4.31)\\
&\le \left\|\mathbf{\Sigma}\mathbf{G}^{-1}\right\|_F + \left\|\mathbf{\Sigma}\right\|_2\left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2\left\|\mathbf{G}^{-1}\right\|_F\\
&\le \left\|\mathbf{\Sigma}\mathbf{G}^{-1}\right\|_F + \epsilon\rho\left\|\mathbf{A}\right\|_2, \qquad (4.32)
\end{aligned}
$$

where Eq. (4.30) follows from Woodbury matrix identity, Eq. (4.31) follows from triangle inequality and Eq. (4.32) follows from Lemma 4.5, the fact that $\mathbf{S}$ is an $(n, \delta, \epsilon/4)$-OSE and the definition that $\rho = \left\|\mathbf{G}^{-1}\right\|_F$.

Now, we can bound $\operatorname{var}(\tilde{\mathbf{b}})$ as follows

$$
\begin{aligned}
\operatorname{var}(\tilde{\mathbf{b}}) &= \frac{\sigma^2}{n}\left\|\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1}\right\|_F^2\\
&\le \frac{\sigma^2}{n}\left(\left\|\mathbf{\Sigma}\mathbf{G}^{-1}\right\|_F + \epsilon\rho\left\|\mathbf{A}\right\|_2\right)^2
\end{aligned}
$$

$$\leq \frac{\sigma^2}{n}\left(\left\|\mathbf{\Sigma G}^{-1}\right\|_F^2 + 2\epsilon\rho\left\|\mathbf{A}\right\|_2\left\|\mathbf{\Sigma G}^{-1}\right\|_F + \epsilon^2\rho^2\left\|\mathbf{A}\right\|_2^2\right)$$

$$\leq \frac{\sigma^2}{n}\left(\left\|\mathbf{\Sigma G}^{-1}\right\|_F^2 + 2\epsilon\rho^2\left\|\mathbf{A}\right\|_2^2 + \epsilon^2\rho^2\left\|\mathbf{A}\right\|_2^2\right) \tag{4.33}$$

$$\leq \mathrm{var}(\mathbf{b}^*) + (2\epsilon+\epsilon^2)\frac{\sigma^2}{n}\rho^2\left\|\mathbf{A}\right\|_2^2, \tag{4.34}$$

where Eq. (4.33) follows from the definition $\rho = \left\|\mathbf{G}^{-1}\right\|_F$.

Next, we bound the bias component bias($\tilde{\mathbf{b}}$). We can simplify bias($\tilde{\mathbf{b}}$) as follows

$$\begin{aligned}
\mathrm{bias}(\tilde{\mathbf{b}}) &= \frac{1}{n}\left\|\mathbf{E}\left[\tilde{\mathbf{b}}\right] - \mathbf{b}_0\right\|^2 \\
&= \frac{1}{n}\left\|\mathbf{AV}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{E}\left[\mathbf{b}\right] - \mathbf{b}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\mathbf{AV}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{Ax}_0 - \mathbf{Ax}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\mathbf{U\Sigma}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{Ax}_0 - \mathbf{U\Sigma V}^T\mathbf{x}_0\right\|_2^2 \\
&= \frac{1}{n}\left\|\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1}\mathbf{\Sigma V}^T\mathbf{x}_0 - \mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2 \\
&= \frac{1}{n}\left\|(\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2^2,
\end{aligned}$$

where we have dropped $\mathbf{U}$ which does not change l_2 norms. Using the same method, bias($\mathbf{b}^*$) can be simplified as

$$\mathrm{bias}(\mathbf{b}^*) = \frac{1}{n}\left\|(\mathbf{\Sigma G}^{-1} - \mathbf{I})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2^2.$$

It is easy to see that $\sqrt{n\,\mathrm{bias}(\tilde{\mathbf{b}})} = \left\|(\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2$ and we can bound it as follows

$$\begin{aligned}
\sqrt{n\,\mathrm{bias}(\tilde{\mathbf{b}})} &= \left\|(\mathbf{\Sigma}\tilde{\mathbf{G}}^{-1} - \mathbf{I})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2 \\
&= \left\|(\mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{G}^{-1} - \mathbf{I})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2 \\
&= \left\|(\mathbf{\Sigma G}^{-1} - \mathbf{I} - \mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{RG}^{-1})\mathbf{\Sigma V}^T\mathbf{x}_0\right\|_2
\end{aligned} \tag{4.35}$$

$$\leq \left\|(\mathbf{\Sigma}\mathbf{G}^{-1} - \mathbf{I})\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 + \left\|\mathbf{\Sigma}(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 \tag{4.36}$$

$$\leq \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \|\mathbf{\Sigma}\|_2 \left\|(\mathbf{I}+\mathbf{R})^{-1}\mathbf{R}\right\|_2 \left\|\mathbf{G}^{-1}\mathbf{\Sigma}\right\|_2 \left\|\mathbf{V}^T\mathbf{x}_0\right\|_2$$

$$\leq \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \|\mathbf{A}\|_2 \cdot \epsilon \cdot \max_i \frac{\sigma_i^2}{\lambda + \sigma_i^2} \|\mathbf{x}_0\|_2 \tag{4.37}$$

$$\leq \sqrt{n\,\mathrm{bias}(\mathbf{b}^*)} + \epsilon \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 , \tag{4.38}$$

where Eq. (4.35) follows from matrix inversion lemma, Eq. (4.36) is the triangle inequality and Eq. (4.37) follows from Lemma 4.5 and that $\mathbf{S}$ is an $(r, \delta, \epsilon/4)$-OSE. Now, dividing both sides of the above inequality Eq. (4.38) by $\sqrt{n}$, we have that

$$\sqrt{\mathrm{bias}(\tilde{\mathbf{b}})} \leq \sqrt{\mathrm{bias}(\mathbf{b}^*)} + \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 .$$

Finally, we obtain the following bound on $\mathrm{bias}(\tilde{\mathbf{b}})$

$$\begin{aligned}
\mathrm{bias}(\tilde{\mathbf{b}}) &\leq \mathrm{bias}(\mathbf{b}^*) + 2\sqrt{\mathrm{bias}(\mathbf{b}^*)} \cdot \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\leq \mathrm{bias}(\mathbf{b}^*) + \frac{2}{\sqrt{n}} \left\|(\mathbf{\Sigma}\mathbf{G}^{-1} - \mathbf{I})\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}_0\right\|_2 \cdot \frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2 \\
&\quad + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\leq \mathrm{bias}(\mathbf{b}^*) + \\
&2\left(\frac{1}{\sqrt{n}} \left\|\mathbf{\Sigma}\mathbf{G}^{-1} - \mathbf{I}\right\|_2 \|\mathbf{\Sigma}\|_2 \left\|\mathbf{V}^T\mathbf{x}_0\right\|_2\right)\left(\frac{\epsilon}{\sqrt{n}} \|\mathbf{A}\|_2 \|\mathbf{x}_0\|_2\right) + \\
&\frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 \\
&\leq \mathrm{bias}(\mathbf{b}^*) + \frac{2\epsilon}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 + \frac{\epsilon^2}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 && (4.39) \\
&= \mathrm{bias}(\mathbf{b}^*) + \frac{(2\epsilon + \epsilon^2)}{n} \|\mathbf{A}\|_2^2 \|\mathbf{x}_0\|_2^2 , && (4.40)
\end{aligned}$$

where Eq. (4.39) holds since $\|\mathbf{\Sigma}\|_2 = \|\mathbf{A}\|_2$, $\left\|\mathbf{V}^T\mathbf{x}_0\right\|_2 = \|\mathbf{x}_0\|_2$ and $\left\|\mathbf{\Sigma}\mathbf{G}^{-1} - \mathbf{I}\right\|_2 = \max_i \frac{\lambda}{\lambda+\sigma_i^2} \le 1$.

The theorem follows immediately from Eq. (4.29), Eq. (4.34), Eq. (4.40) and the fact that $\epsilon < 1$. □

4.10 Proof of Theorem 4.4

We first prove a generalization of Lemma 4.1 as follows.

Lemma 4.7. *Given a full rank matrix* $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$ *and* $\lambda > 0$. *Suppose that* $\mathbf{S} \in \mathbb{R}^{t\times p}$ *is an* $(n, \delta, \epsilon/4)$*-OSE for* $\epsilon \in (0,1)$ *and* $\delta \in (0,1)$. *Then, with probability* $1-\delta$, *we have*

$$\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F \le \epsilon \left\|\mathbf{X}^*\right\|_F,$$

where $\tilde{\mathbf{X}}$ *is given by Eq.* (4.9) *and* $\mathbf{X}^*$ *is the optimal solution to multiple response ridge regression Eq.* (4.7).

Proof. Let the thin SVD of $\mathbf{A}$ be $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Fix an arbitrary $i \in [m]$, consider the column vectors $\tilde{\mathbf{X}}^{(i)}$ and $\mathbf{X}^{*(i)}$. By definition, we can see that $\mathbf{X}^{*(i)} = \mathbf{A}^T(\lambda\mathbf{I}_n + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{B}^{(i)}$ and

$$\tilde{\mathbf{X}}^{(i)} = \mathbf{A}^T(\mathbf{A}\mathbf{S}^T)^{\dagger^T}(\lambda(\mathbf{A}\mathbf{S}^T)^{\dagger^T} + \mathbf{A}\mathbf{S}^T)^{\dagger}\mathbf{B}^{(i)}.$$

Define $\tilde{\mathbf{G}} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}(\mathbf{S}\mathbf{V})^T(\mathbf{S}\mathbf{V})$ and $\mathbf{G} = \lambda\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}$. Now, we can regard $\mathbf{B}^{(i)}$ as the target vector and then apply Lemma 4.3 on $\mathbf{X}^{*(i)}$ and Lemma 4.4 on $\tilde{\mathbf{X}}^{(i)}$, respectively. This shows that $\mathbf{X}^{*(i)} = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}^{(i)}$ and $\tilde{\mathbf{X}}^{(i)} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{B}^{(i)}$. Hence, combining all columns $i \in [m]$, we have that $\mathbf{X}^* = \mathbf{V}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B}$ and $\tilde{\mathbf{X}} = \mathbf{V}\tilde{\mathbf{G}}^{-1}\mathbf{U}^T\mathbf{B}$.

Similar to the proof of Lemma 4.1, we recall the definition $\mathbf{R} = \mathbf{G}^{-1}(\tilde{\mathbf{G}} - \mathbf{G})$ and write $\tilde{\mathbf{G}}^{-1} = (\mathbf{I} + \mathbf{R})^{-1}\mathbf{G}^{-1}$. Let $\epsilon' = \epsilon/4$. Then, we have

$$\left\|\tilde{\mathbf{X}} - \mathbf{X}^*\right\|_F = \left\|\mathbf{V}(\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{B}\right\|_F$$

$$
\begin{aligned}
&= \left\| (\tilde{\mathbf{G}}^{-1} - \mathbf{G}^{-1})\mathbf{U}^T\mathbf{B} \right\|_F \\
&= \left\| \left((\mathbf{I} + \mathbf{R})^{-1} - \mathbf{I} \right) \mathbf{G}^{-1}\mathbf{U}^T\mathbf{B} \right\|_F \\
&= \left\| (\mathbf{I} + \mathbf{R})^{-1}\mathbf{R}\mathbf{G}^{-1}\mathbf{U}^T\mathbf{B} \right\|_F && (4.41) \\
&\leq \left\| (\mathbf{I} + \mathbf{R})^{-1}\mathbf{R} \right\|_2 \left\| \mathbf{G}^{-1}\mathbf{U}^T\mathbf{B} \right\|_F \\
&= \left\| (\mathbf{I} + \mathbf{R})^{-1}\mathbf{R} \right\|_2 \left\| \mathbf{X}^* \right\|_F \\
&\leq \frac{2\epsilon' + \epsilon'^2}{1 - (2\epsilon' + \epsilon'^2)} \left\| \mathbf{X}^* \right\|_F && (4.42) \\
&\leq 4\epsilon' \left\| \mathbf{X}^* \right\|_F \\
&= \epsilon \left\| \mathbf{X}^* \right\|_F . && (4.43)
\end{aligned}
$$

where Eq. (4.41) follows from matrix inversion lemma and Eq. (4.42) follows from Lemma 4.5 and the fact that $\mathbf{S}$ is an $(r, \delta, \epsilon/4)$-OSE. □

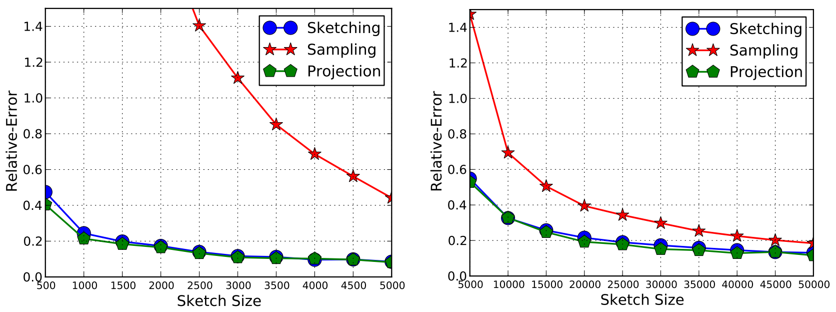
Now, Theorem 4.4 restated below is an immediate consequence of Lemma 4.7.

Theorem 4.4. *Given a full rank matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\lambda > 0$, parameter $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. Select integers t', t such that $t' \geq 2\delta^{-1}(n^2 + n)/(\epsilon/6 - \epsilon^2/144)^2$ and $t \geq 72\epsilon^{-1}[\sqrt{n} + \sqrt{8 \log(6p/\delta)}]^2 \log(6n/\delta)$. Let $\mathbf{S} = \mathbf{\Phi}_{srht}\mathbf{\Phi}_{sparse}$, where $\mathbf{\Phi}_{sparse} \in \mathbb{R}^{t' \times p}$ is a sparse embedding matrix and $\mathbf{\Phi}_{srht} \in \mathbb{R}^{t \times t'}$ is an SRHT matrix. Then, with probability at least $1 - \delta$, we have*

$$\left\| \tilde{\mathbf{X}} - \mathbf{X}^* \right\|_F \leq \epsilon \left\| \mathbf{X}^* \right\|_F ,$$

where $\tilde{\mathbf{X}}$ is given by Eq. (4.9) and $\mathbf{X}^*$ is the optimal solution to multiple response ridge regression Eq. (4.7). In addition, the total time complexity of computing $\mathbf{A}\mathbf{S}^T$ and $\tilde{\mathbf{X}}$ is $O(\mathrm{nnz}(\mathbf{A}) + n^3 \log(\frac{n}{\epsilon})/\epsilon^2 + nm)$.

Proof of Theorem 4.4. The bound on $\left\| \tilde{\mathbf{X}} - \mathbf{X} \right\|_F$ follows immediately from Lemma 4.7 and Theorem 4.1 which shows that



(a) ARCENE: Relative-error (b) DOROTHEA: Relative-Error

Figure 4.3: Relative-error of competing algorithms on realworld datasets

$\mathbf{S} = \mathbf{\Phi}_{\text{srht}}\mathbf{\Phi}_{\text{sparse}}$ is an $(n, \delta, \epsilon/4)$-OSE. And the running time analysis is similar to the one given in Section 4.2. □

4.11 Experimental Results: Relative-error on Realworld Datasets

In this section, we present the deferred experiment results of relative-error of competing baselines on datasets ARCENE and DOROTHEA. The experiment configuration is the same to Section 4.5.2. Figure 4.3 plots the results. We see that, similar to the results on the synthetic dataset, the relative-error decreases as t increases. In addition, the SKETCHING algorithm and the PROJECTION algorithm outperform the SAMPLING algorithm in terms of accuracy on both datasets. Notice that, for moderately large t, the SKETCHING algorithm achieves a relative-error that is smaller than 20% on both datasets.

Chapter 5

Recovery of Pairwise Interaction Tensors

Tensor completion from incomplete observations is a problem of significant practical interest. However, it is unlikely that there exists an efficient algorithm with provable guarantee to recover a general tensor from a limited number of observations. In this chapter, we study the recovery algorithm for pairwise interaction tensors, which has recently gained considerable attention for modeling multiple attribute data due to its simplicity and effectiveness. Specifically, in the absence of noise, we show that one can exactly recover a pairwise interaction tensor by solving a constrained convex program which minimizes the weighted sum of nuclear norms of matrices from $O(nr\log^2(n))$ observations. For the noisy cases, we also prove error bounds for a constrained convex program for recovering the tensors. Our experiments on the synthetic dataset demonstrate that the recovery performance of our algorithm agrees well with the theory. In addition, we apply our algorithm on a temporal collaborative filtering task and obtain state-of-the-art results.

5.1 Introduction

Many tasks of recommender systems can be formulated as recovering an unknown tensor (multi-way array) from a few observations of its entries [122, 152, 151, 135]. Recently, convex optimization algorithms for recovering a matrix, which is a special case of tensor, have been extensively studied [56, 142, 54]. Moreover, there are several theoretical developments that guarantee *exact recovery* of most low-rank matrices from partial observations using nuclear norm minimization [68, 32]. These results seem to suggest a promising direction to solve the general problem of tensor recovery.

However, there are inevitable obstacles to generalize the techniques for matrix completion to tensor recovery, since a number of fundamental computational problems of matrix is NP-hard in their tensorial analogues [73]. For instance, Håstad showed that it is NP-hard to compute the rank of a given tensor [71]; Hillar and Lim proved the NP-hardness to decompose a given tensor into sum of rank-one tensors even if a tensor is fully observed [73]. The existing evidence suggests that it is very unlikely that there exists an efficient exact recovery algorithm for general tensors with missing entries. Therefore, it is natural to ask whether it is possible to identify a useful class of tensors for which we can devise an exact recovery algorithm.

In this chapter, we focus on *pairwise interaction tensors*, which have recently demonstrated strong performance in several recommendation applications, e.g. tag recommendation [121] and sequential data analysis [123]. Pairwise interaction tensors are a special class of general tensors, which directly model the pairwise interactions between different attributes. Take movie recommendation as an example, to model a user's ratings for movies varying over time, a pairwise interaction tensor assumes that each rating is determined by three factors: the user's inherent preference on the movie, the movie's trending popularity and the user's varying mood over time. Formally, pairwise interaction tensor assumes that each entry T_{ijk} of a tensor $\mathcal{T}$ of size $n_1 \times n_2 \times n_3$ is given by following

$$T_{ijk} = \left\langle \mathbf{u}_i^{(a)}, \mathbf{v}_j^{(a)} \right\rangle + \left\langle \mathbf{u}_j^{(b)}, \mathbf{v}_k^{(b)} \right\rangle + \left\langle \mathbf{u}_k^{(c)}, \mathbf{v}_i^{(c)} \right\rangle, \quad \text{for all } (i,j,k) \in [n_1] \times [n_2] \times [n_3], \tag{5.1}$$

where $\{\mathbf{u}_i^{(a)}\}_{i\in[n_1]}$, $\{\mathbf{v}_i^{(a)}\}_{j\in[n_2]}$ are r_1 dimensional vectors, $\{\mathbf{u}_j^{(b)}\}_{j\in[n_2]}$, $\{\mathbf{v}_k^{(b)}\}_{k\in[n_3]}$ are r_2 dimensional vectors and $\{\mathbf{u}_k^{(c)}\}_{k\in[n_3]}$, $\{\mathbf{v}_i^{(c)}\}_{i\in[n_1]}$ are r_3 dimensional vectors, respectively. [1]

The existing recovery algorithms for pairwise interaction tensor use local optimization methods, which do not guarantee the recovery performance [123, 121]. In this chapter, we design efficient recovery algorithms for pairwise interaction tensors with rigorous guarantee. More specifically, in the absence of noise, we show that one can exactly recover a pairwise interaction tensor by solving a constrained convex program which minimizes the weighted sum of nuclear norms of matrices from $O(nr\log^2(n))$ observations, where $n = \max\{n_1, n_2, n_3\}$ and $r = \max\{r_1, r_2, r_3\}$. For noisy cases, we also prove error bounds for a constrained convex program for recovering the tensors.

In the proof of our main results, we reformulated the recovery problem as a constrained matrix completion problem with a special observation operator. Previously, Gross et al. [68] have showed that the nuclear norm heuristic can exactly recover low rank matrix from a sufficient number of observations of an orthogonal observation operator. We note that the orthogonality is critical to their argument. However, the observation operator, in our case, turns out to be non-orthogonal, which becomes a major challenge in our proof. In order to deal with the non-orthogonal operator, we have substantially extended their technique in our

[1] For simplicity, we only consider three-way tensors in this chapter.

proof. We believe that our technique can be generalized to handle other matrix completion problem with non-orthogonal observation operators.

Moreover, we extend existing singular value thresholding method to develop a simple and scalable algorithm for solving the recovery problem in both exact and noisy cases. Our experiments on the synthetic dataset demonstrate that the recovery performance of our algorithm agrees well with the theory. Finally, we apply our algorithm on a temporal collaborative filtering task and obtain state-of-the-art results.

5.2 Recovering pairwise interaction tensors

In this section, we first introduce the matrix formulation of pairwise interaction tensors and specify the recovery problem. Then we discuss the sufficient conditions on pairwise interaction tensors for which an exact recovery would be possible. After that we formulate the convex program for solving the recovery problem and present our theoretical results on the sample bounds for achieving an exact recovery. In addition, we also show a quadratically constrained convex program is stable for the recovery from noisy observations.

A matrix formulation of pairwise interaction tensors. The original formulation of pairwise interaction tensors by Rendle et al. [121] is given by Eq. (5.1), in which each entry of a tensor is the sum of inner products of feature vectors. We can reformulate Eq. (5.1) more concisely using matrix notations. In particular, we can rewrite Eq. (5.1) as follows

$$T_{ijk} = A_{ij} + B_{jk} + C_{ki}, \quad \text{for all } (i,j,k) \in [n_1] \times [n_2] \times [n_3], \tag{5.2}$$

where we set $A_{ij} = \left\langle \mathbf{u}_i^{(a)}, \mathbf{v}_j^{(a)} \right\rangle$, $B_{jk} = \left\langle \mathbf{u}_j^{(b)}, \mathbf{v}_k^{(b)} \right\rangle$, and $C_{ki} = \left\langle \mathbf{u}_k^{(c)}, \mathbf{v}_i^{(c)} \right\rangle$ for all (i,j,k). Clearly, matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are rank

r_1, r_2 and r_3 matrices, respectively.

We call tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ a *pairwise interaction tensor*, which is denoted as $\mathcal{T} = \text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, if $\mathcal{T}$ obeys Eq. (5.2). We note that this concise definition is equivalent to the original one. In the rest of this chapter, we will exclusively use the matrix formulation of pairwise interaction tensors.

Recovery problem. Suppose we have partial observations of a pairwise interaction tensor $\mathcal{T} = \text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$. We write $\Omega \subseteq [n_1] \times [n_2] \times [n_3]$ to be the set of indices of m observed entries. In this work, we shall assume Ω is sampled uniformly from the collection of all sets of size m. Our goal is to recover matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and therefore the entire tensor $\mathcal{T}$ from exact or noisy observations of $\{T_{ijk}\}_{(ijk) \in \Omega}$.

Before we proceed to the recovery algorithm, we first discuss when the recovery is possible.

Recoverability: uniqueness. The original recovery problem for pairwise interaction tensors is ill-posed due to a uniqueness issue. In fact, for any pairwise interaction tensor $\mathcal{T} = \text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, we can construct infinitely manly different sets of matrices $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ such that $\text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \text{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$. For example, we have $T_{ijk} = A_{ij} + B_{jk} + C_{ki} = (A_{ij} + \delta a_i) + B_{jk} + (C_{ki} + (1-\delta) a_i)$, where $\delta \neq 0$ can be any non-zero constant and $\mathbf{a}$ is an arbitrary non-zero vector of size n_1. Now, we can construct $\mathbf{A}', \mathbf{B}'$ and $\mathbf{C}'$ by setting $A'_{ij} = A_{ij} + \delta a_i, B'_{jk} = B_{jk}$ and $C'_{ki} = C_{ki} + (1 - \delta) a_i$. It is clear that $\mathcal{T} = \text{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$.

This ambiguity prevents us to recover $\mathbf{A}, \mathbf{B}, \mathbf{C}$ even if $\mathcal{T}$ is fully observed, since it is entirely possible to recover $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ instead of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ based on the observations. In order to avoid this obstacle, we construct a set of constraints such that, given any pairwise interaction tensor $\text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, there exists unique matrices $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ satisfying the constraints and obeys $\text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \text{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$. Formally, we prove the following proposition.

Proposition 5.1. *For any pairwise interaction tensor* $\mathcal{T} =$

$\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, *there exists unique* $\mathbf{A}' \in S_A, \mathbf{B}' \in S_B, \mathbf{C}' \in S_C$ *such that* $\mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{Pair}(\mathbf{A}', \mathbf{B}', \mathbf{C}')$ *where we define* $S_B = \{\mathbf{M} \in \mathbb{R}^{n_2 \times n_3} : \mathbf{1}^T\mathbf{M} = \mathbf{0}^T\}$,$S_C = \{\mathbf{M} \in \mathbb{R}^{n_3 \times n_1} : \mathbf{1}^T\mathbf{M} = \mathbf{0}^T\}$ *and* $S_A = \{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2} : \mathbf{1}^T\mathbf{M} = \left(\frac{1}{n_2}\mathbf{1}^T\mathbf{M}\mathbf{1}\right)\mathbf{1}^T\}$.

We point out that there is a natural connection between the uniqueness issue and the "bias" components, which is a quantity of much attention in the field of recommender system [88]. Due to lack of space, we defer the detailed discussion on this connection and the proof of Proposition 5.1 to Section 5.8.

Recoverability: incoherence. It is easy to see that recovering a pairwise tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{0}, \mathbf{0})$ is equivalent to recover the matrix $\mathbf{A}$ from a subset of its entries. Therefore, the recovery problem of pairwise interaction tensors subsumes matrix completion problem as a special case. Previous studies have confirmed that the *incoherence condition* is an essential requirement on the matrix in order to guarantee a successful recovery of matrices. This condition can be stated as follows.

Let $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$ be the singular value decomposition of a rank r matrix $\mathbf{M}$. We call matrix $\mathbf{M}$ is (μ_0, μ_1)-incoherent if $\mathbf{M}$ satisfies:

A0. For all $i \in [n_1]$ and $j \in [n_2]$, we have $\frac{n_1}{r}\sum_{k\in[r]} U_{ik}^2 \leq \mu_0$ and $\frac{n_2}{r}\sum_{k\in[r]} V_{jk}^2 \leq \mu_0$.

A1. The maximum entry of $\mathbf{U}\mathbf{V}^T$ is bounded by $\mu_1\sqrt{r/(n_1 n_2)}$ in absolute value.

It is well known the recovery is possible only if the matrix is (μ_0, μ_1)-incoherent for bounded μ_0, μ_1 (i.e, μ_0, μ_1 is polylogarithmic with respect to n). Since the matrix completion problem is reducible to the recovery problem for pairwise interaction tensors, our theoretical result will inherit the incoherence assumptions on matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

Exact recovery in the absence of noise. We first consider the scenario where the observations are exact. Specifically, suppose we are given m observations $\{T_{ijk}\}_{(ijk)\in\Omega}$, where Ω is sampled from uniformly at random from $[n_1] \times [n_2] \times [n_3]$. We propose to recover matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and therefore tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ using the following convex program,

$$\begin{aligned} \underset{\mathbf{X}\in S_A, \mathbf{Y}\in S_B, \mathbf{Z}\in S_C}{\text{minimize}} \quad & \sqrt{n_3}\left\|\mathbf{X}\right\|_* + \sqrt{n_1}\left\|\mathbf{Y}\right\|_* + \sqrt{n_2}\left\|\mathbf{Z}\right\|_* \\ \text{subject to} \quad & X_{ij} + Y_{jk} + Z_{ki} = T_{ijk}, \quad (i,j,k) \in \Omega, \end{aligned} \tag{5.3}$$

where $\left\|\mathbf{M}\right\|_*$ denotes the nuclear norm of matrix $\mathbf{M}$, which is the sum of singular values of $\mathbf{M}$, and S_A, S_B, S_C is defined in Proposition 5.1.

We show that, under the incoherence conditions, the above nuclear norm minimization method successful recovers a pairwise interaction tensor $\mathcal{T}$ when the number of observations m is $O(nr\log^2 n)$ with high probability.

Theorem 5.1. *Let $\mathcal{T} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a pairwise interaction tensor $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$ as defined in Proposition 5.1. Without loss of generality assume that $9 \le n_1 \le n_2 \le n_3$. Suppose we observed m entries of $\mathcal{T}$ with the locations sampled uniformly at random from $[n_1]\times[n_2]\times[n_3]$ and also suppose that each of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ is (μ_0, μ_1)-incoherent. Then, there exists a universal constant C, such that if*

$$m > C\max\{\mu_1^2, \mu_0\}n_3 r\beta\log^2(6n_3),$$

where $r = \max\{\mathrm{rank}(\mathbf{A}), \mathrm{rank}(\mathbf{B}), \mathrm{rank}(\mathbf{C})\}$ and $\beta > 2$ is a parameter, the minimizing solution $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ for program Eq. (5.3) is unique and satisfies $\mathbf{X} = \mathbf{A}, \mathbf{Y} = \mathbf{B}, \mathbf{Z} = \mathbf{C}$ with probability at least $1 - \log(6n_3)6n_3^{2-\beta} - 3n_3^{2-\beta}$.

Stable recovery in the presence of noise. Now, we move to the case where the observations are perturbed by noise with bounded energy. In particular, our noisy model assumes that we observe

$$\hat{T}_{ijk} = T_{ijk} + \sigma_{ijk}, \quad \text{for all } (i,j,k) \in \Omega, \tag{5.4}$$

where σ_{ijk} is a noise term, which maybe deterministic or stochastic. We assume σ has bounded energy on Ω and specifically that $\|\mathcal{P}_\Omega(\sigma)\|_F \leq \epsilon_1$ for some $\epsilon_1 > 0$, where $\mathcal{P}_\Omega(\cdot)$ denotes the restriction on Ω. Under this assumption on the observations, we derive the error bound of the following quadratically-constrained convex program, which recover $\mathcal{T}$ from the noisy observations.

$$\begin{aligned} \underset{\mathbf{X}\in S_A,\mathbf{Y}\in S_B,\mathbf{Z}\in S_C}{\text{minimize}} \quad & \sqrt{n_3}\,\|\mathbf{X}\|_* + \sqrt{n_1}\,\|\mathbf{Y}\|_* + \sqrt{n_2}\,\|\mathbf{Z}\|_* \\ \text{subject to} \quad & \left\|\mathcal{P}_\Omega(\mathrm{Pair}(\mathbf{X},\mathbf{Y},\mathbf{Z})) - \mathcal{P}_\Omega(\hat{\mathcal{T}})\right\|_F \leq \epsilon_2. \end{aligned} \tag{5.5}$$

Theorem 5.2. *Let* $\mathcal{T} = \mathrm{Pair}(\mathbf{A},\mathbf{B},\mathbf{C})$ *and* $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$*. Let* Ω *be the set of observations as described in Theorem 5.1. Suppose we observe* $\hat{T}_{ijk}$ *for* $(i,j,k) \in \Omega$ *as defined in Eq.* (5.4) *and also assume that* $\|\mathcal{P}_\Omega(\sigma)\|_F \leq \epsilon_1$ *holds. Denote the reconstruction error of the optimal solution* $\mathbf{X},\mathbf{Y},\mathbf{Z}$ *of convex program Eq.* (5.5) *as* $\mathbf{E} = \mathrm{Pair}(\mathbf{X},\mathbf{Y},\mathbf{Z}) - \mathcal{T}$*. Also assume that* $\epsilon_1 \leq \epsilon_2$*. Then, we have*

$$\|\mathbf{E}\|_* \leq 5\sqrt{\frac{2rn_1n_2^2}{8\beta\log(n_1)}}(\epsilon_1+\epsilon_2),$$

with probability at least $1 - \log(6n_3)6n_3^{2-\beta} - 3n_3^{2-\beta}$.

The proof of Theorem 5.1 and Theorem 5.2 is available in Section 5.6 and Section 5.7.

Related work. Rendle et al. [121] proposed pairwise interaction tensors as a model used for tag recommendation. In a subsequent work, Rendle et al. [123] applied pairwise interaction tensors in the sequential analysis of purchase data. In both applications, their methods using pairwise interaction tensor demonstrated excellent performance. However, their algorithms are prone to local optimal issues and the recovered tensor might be very different from its true value. In contrast, our main results, Theorem 5.1 and Theorem 5.2, guarantee that a convex program can exactly or accurately recover the pairwise interaction tensors from $O(nr\log^2(n))$ observations. In this sense, our work can be considered as a more effective way to recover pairwise interaction tensors from partial observations.

In practice, various tensor factorization methods are used for estimating missing entries of tensors [85, 134, 3, 152, 115]. In addition, inspired by the success of nuclear norm minimization heuristics in matrix completion, several work used a generalized nuclear norm for tensor recovery [147, 148, 98]. However, these work do not guarantee exact recovery of tensors from partial observations.

5.3 Scalable optimization algorithm

There are several possible methods to solving the optimization problems Eq. (5.3) and Eq. (5.5). For small problem sizes, one may reformulate the optimization problems as semi-definite programs and solve them using interior point method. The state-of-the-art interior point solvers offer excellent accuracy for finding the optimal solution. However, these solvers become prohibitively slow for pairwise interaction tensors larger than 100 × 100 × 100. In order to apply the recover algorithms on large scale pairwise interaction tensors, we use singular value thresholding (SVT) algorithm proposed recently by Cai et al. [30], which is a first-order method with promising performance for solving nuclear norm minimization problems.

We first discuss the SVT algorithm for solving the exact completion problem Eq. (5.3). For convenience, we reformulate the original optimization objective Eq. (5.3) as follows,

$$\begin{aligned} &\underset{\mathbf{X}\in S_A,\mathbf{Y}\in S_B,\mathbf{Z}\in S_C}{\text{minimize}} && \|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_* \\ &\text{subject to} && \frac{X_{ij}}{\sqrt{n_3}} + \frac{Y_{jk}}{\sqrt{n_1}} + \frac{Z_{ki}}{\sqrt{n_2}} = T_{ijk}, \quad (i,j,k)\in\Omega, \end{aligned} \tag{5.6}$$

where we have incorporated coefficients on the nuclear norm terms into the constraints. It is easy to see that the recovered tensor is given by $\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})$, where $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ is the optimal solution of Eq. (5.6). Our algorithm solves a slightly relaxed version of the reformulated objective Eq. (5.6),

$$\underset{\mathbf{X}\in S_A, \mathbf{Y}\in S_B, \mathbf{Z}\in S_C}{\text{minimize}} \tau\left(\|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_*\right) + \frac{1}{2}\left(\|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2 + \|\mathbf{Z}\|_F^2\right) \tag{5.7}$$

$$\text{subject to} \quad \frac{X_{ij}}{\sqrt{n_3}} + \frac{Y_{jk}}{\sqrt{n_1}} + \frac{Z_{ki}}{\sqrt{n_2}} = T_{ijk}, \quad (i,j,k) \in \Omega.$$

It is easy to see that Eq. (5.7) is closely related to Eq. (5.6) and the original problem Eq. (5.3), as the relaxed problem converges to the original one as $\tau \to \infty$. Therefore by selecting a large value the parameter τ, a minimizing solution to Eq. (5.7) nearly minimizes Eq. (5.3).

Our algorithm iteratively minimizes Eq. (5.7) and produces a sequence of matrices $\{\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k\}$ converging to the optimal solution $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ that minimizes Eq. (5.7). We begin with several definitions. For observations $\Omega = \{a_i, b_i, c_i | i \in [m]\}$, let operators $\mathcal{P}_{\Omega_A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$, $\mathcal{P}_{\Omega_B} : \mathbb{R}^{n_2\times n_3} \to \mathbb{R}^m$ and $\mathcal{P}_{\Omega_C} : \mathbb{R}^{n_3\times n_1} \to \mathbb{R}^m$ represents the influence of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ on the m observations. In particular,

$$\mathcal{P}_{\Omega_A}(\mathbf{X}) = \frac{1}{\sqrt{n_3}}\sum_{i=1}^m X_{a_i b_i}\delta_i, \quad \mathcal{P}_{\Omega_B}(\mathbf{Y}) = \frac{1}{\sqrt{n_1}}\sum_{i=1}^m Y_{b_i c_i}\delta_i, \text{ and}$$

$$\mathcal{P}_{\Omega_C}(\mathbf{Z}) = \frac{1}{\sqrt{n_2}}\sum_{i=1}^m Z_{c_i a_i}\delta_i.$$

It is easy to verify that

$$\mathcal{P}_{\Omega_A}(\mathbf{X}) + \mathcal{P}_{\Omega_B}(\mathbf{Y}) + \mathcal{P}_{\Omega_C}(\mathbf{Z}) = \mathcal{P}_{\Omega}(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})).$$

We also denote $\mathcal{P}_{\Omega_A}^*$ be the adjoint operator of $\mathcal{P}_{\Omega_A}$ and similarly define $\mathcal{P}_{\Omega_B}^*$ and $\mathcal{P}_{\Omega_C}^*$. Finally, for a matrix $\mathbf{X}$ for size $n_1 \times$

n_2, we define $\text{center}(\mathbf{X}) = \mathbf{X} - \frac{1}{n_1}\mathbf{1}\mathbf{1}^T\mathbf{X}$ as the column centering operator that removes the mean of each n_2 columns, i.e., $\mathbf{1}^T\text{center}(\mathbf{X}) = \mathbf{0}^T$.

Starting with $\mathbf{y}^0 = 0$ and $k = 1$, our algorithm iteratively computes

$$\begin{aligned}
\text{Step (1).} \quad \mathbf{X}^k &= \text{shrink}_A(\mathcal{P}^*_{\Omega_A}(\mathbf{y}^{k-1}), \tau),\\
\mathbf{Y}^k &= \text{shrink}_B(\mathcal{P}^*_{\Omega_B}(\mathbf{y}^{k-1}), \tau),\\
\mathbf{Z}^k &= \text{shrink}_C(\mathcal{P}^*_{\Omega_C}(\mathbf{y}^{k-1}), \tau),\\
\text{Step (2e).} \quad \mathbf{e}^k &= \mathcal{P}_\Omega(\mathcal{T}) - \mathcal{P}_\Omega(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}))\\
\mathbf{y}^k &= \mathbf{y}^{k-1} + \delta\mathbf{e}^k.
\end{aligned}$$

Here shrink_A is a shrinkage operator defined as follows

$$\text{shrink}_A(\mathbf{M}, \tau) \triangleq \arg\min_{\tilde{\mathbf{M}}\in S_A} \frac{1}{2}\left\|\tilde{\mathbf{M}} - \mathbf{M}\right\|_F^2 + \tau\left\|\tilde{\mathbf{M}}\right\|_*. \tag{5.8}$$

Shrinkage operators shrink_B and shrink_C are defined similarly except they require $\tilde{\mathbf{M}}$ belongs S_B and S_C, respectively. We note that our definition of the shrinkage operators shrink_A, shrink_B and shrink_C are slightly different from that of the original SVT [30] algorithm, where $\tilde{\mathbf{M}}$ is unconstrained. We can show that our constrained version of shrinkage operators can also be calculated using singular value decompositions of column centered matrices.

Let the SVD of the column centered matrix $\text{center}(\mathbf{M})$ be $\text{center}(\mathbf{M}) = \mathbf{U}\Sigma\mathbf{V}^T, \quad \Sigma = \text{diag}(\{\sigma_i\})$. We can prove that the shrinkage operator shrink_B is given by

$$\text{shrink}_B(\mathbf{M}, \tau) = \mathbf{U}\,\text{diag}(\{\sigma_i - \tau\}_+)\mathbf{V}^T, \tag{5.9}$$

where s_+ is the positive part of s, that is, $s_+ = \max\{0, s\}$. Since subspace S_C is structurally identical to S_B, it is easy to see that the calculation of shrink_C is identical to that of shrink_B. The computation of shrink_A is a little more complicated. We have

$$\text{shrink}_A(\mathbf{M},\tau) = \mathbf{U}\,\text{diag}(\{\sigma_i-\tau\}_+)\mathbf{V}^T+\frac{1}{\sqrt{n_1n_2}}\left(\{\delta-\tau\}_+ + \{\delta+\tau\}_-\right)\mathbf{1}\mathbf{1}^T, \tag{5.10}$$

where $\mathbf{U}\Sigma\mathbf{V}^T$ is still the SVD of $\text{center}(\mathbf{M})$, $\delta = \frac{1}{\sqrt{n_1n_2}}\mathbf{1}^T\mathbf{M}\mathbf{1}$ is a constant and $s_- = \min\{0,s\}$ is the negative part of s. The algorithm iterates between Step (1) and Step (2e) and produces a series of $(\mathbf{X}^k,\mathbf{Y}^k,\mathbf{Z}^k)$ converging to the optimal solution of Eq. (5.7). The iterative procedure terminates when the training error is small enough, namely, $\left\|\mathbf{e}^k\right\|_F \le \epsilon$. We refer interested readers to [30] for a convergence proof of the SVT algorithm.

The optimization problem for noisy completion Eq. (5.5) can be solved in a similar manner. We only need to modify Step (2e) to incorporate the quadratical constraint of Eq. (5.5) as follows

Step (2n). $\quad \mathbf{e}^k = \mathcal{P}_\Omega(\hat{\mathcal{T}}) - \mathcal{P}_\Omega(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}))$

$$\begin{bmatrix}\mathbf{y}^k\\ s^k\end{bmatrix} = \mathcal{P}_\mathcal{K}\left(\begin{bmatrix}\mathbf{y}^{k-1}\\ s^{k-1}\end{bmatrix} + \delta\begin{bmatrix}\mathbf{e}^k\\ -\epsilon\end{bmatrix}\right),$$

where $\mathcal{P}_\Omega(\hat{\mathcal{T}})$ is the noisy observations and the cone projection operator $\mathcal{P}_\mathcal{K}$ can be explicitly computed by

$$\mathcal{P}_\mathcal{K} : (x,t) \to \begin{cases} (x,t) & \text{if } \|x\| \le t, \\ \frac{\|x\|+t}{2\|x\|}(x,\|x\|) & \text{if } -\|x\| \le t \le \|x\|, \\ (0,0) & \text{if } t \le -\|x\|. \end{cases}$$

By iterating between Step (1) and Step (2n) and selecting a sufficiently large τ, the algorithm generates a sequence of $\{\mathbf{X}^k,\mathbf{Y}^k,\mathbf{Z}^k\}$ that converges to a nearly optimal solution to the noisy completion program Eq. (5.5) [30]. We have also included a detailed description of both algorithms in Section 5.9.

At each iteration, we need to compute one singular value decomposition and perform a few elementary matrix additions.

We can see that for each iteration k, $\mathbf{X}^k$ vanishes outside of $\Omega_A = \{a_i b_i\}$ and is sparse. Similarly $\mathbf{Y}^k$,$\mathbf{Z}^k$ are also sparse matrices. Previously, we showed that the computation of shrinkage operators requires a SVD of a column centered matrix center$(\mathbf{M}) - \frac{1}{n_1}\mathbf{1}\mathbf{1}^T\mathbf{X}$, which is the sum of a sparse matrix $\mathbf{M}$ and a rank-one matrix. Clearly the matrix-vector multiplication of the form center$(\mathbf{M})\mathbf{v}$ can be computed with time $O(n+m)$. This enables the use of Lanczos method based SVD implementations for example PROPACK [92] and SVDPACKC [20], which only needs subroutine of calculating matrix-vector products. In our implementation, we develop a customized version of SVDPACKC for computing the shrinkage operators. Further, for an appropriate choice of τ, $\{\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k\}$ turned out to be low rank matrices, which matches the observations in the original SVT algorithm [30]. Hence, the storage cost $\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k$ can be kept low and we only need to perform a partial SVD to get the first r singular vectors. The estimated rank r is gradually increased during the iterations using a similar method suggested in [30, Section 5.1.1]. We can see that, in sum, the overall complexity per iteration of the recovery algorithm is $O(r(n+m))$.

5.4 Experiments

Phase transition in exact recovery. We investigate how the number of measurements affects the success of exact recovery. In this simulation, we fixed $n_1 = 100, n_2 = 150, n_3 = 200$ and $r_1 = r_2 = r_3 = r$. We tested a variety of choices of (r, m) and for each choice of (r, m), we repeat the procedure for 10 times. At each time, we randomly generated $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$ of rank r. We generated $\mathbf{A} \in S_A$ by sampling two factor matrices $\mathbf{U}_A \in \mathbb{R}^{n_1 \times r}, \mathbf{V}_A \in \mathbb{R}^{n_2 \times r}$ with i.i.d. standard Gaussian entries and setting $\mathbf{A} = \mathcal{P}_{S_A}(\mathbf{U}_A \mathbf{V}_A^T)$, where $\mathcal{P}_{S_A}$ is the orthogonal projection onto subspace S_A. Matrices $\mathbf{B} \in S_B$ and

$\mathbf{C} \in S_C$ are sampled in a similar way. We uniformly sampled a subset Ω of m entries and reveal them to the recovery algorithm. We deemed $\mathbf{A}, \mathbf{B}, \mathbf{C}$ successfully recovered if $(\|\mathbf{A}\|_F + \|\mathbf{B}\|_F + \|\mathbf{C}\|_F)^{-1}(\|\mathbf{X} - \mathbf{A}\|_F + \|\mathbf{Y} - \mathbf{B}\|_F + \|\mathbf{Z} - \mathbf{C}\|_F) \leq 10^{-3}$, where $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ are the recovered matrices. Finally, we set the parameters τ, δ of the exact recovery algorithm by $\tau = 10\sqrt{n_1 n_2 n_3}$ and $\delta = 0.9m(n_1 n_2 n_3)^{-1}$.

Figure 5.1 shows the results of these experiments. The x-axis is the ratio between the number of measurements m and the degree of freedom $d = r(n_1 + n_2 - r) + r(n_2 + n_3 - r) + r(n_3 + n_1 - r)$. Note that a value of x-axis smaller than one corresponds to a case where there is infinite number of solutions satisfying given entries. The y-axis is the rank r of the synthetic matrices. The color of each grid indicates the empirical success rate. White denotes exact recovery in all 10 experiments, and black denotes failure for all experiments. From Figure 5.1 (Left), we can see that the algorithm succeeded almost certainly when the number of measurements is 2.5 times or larger than the degree of freedom for most parameter settings. We also observe that, near the boundary of $m/d \approx 2.5$, there is a relatively sharp phase transition. To verify this phenomenon, we repeated the experiments, but only vary m/d between 1.5 and 3.0 with finer steps. The results on Figure 5.1 (Right) shows that the phase transition continued to be sharp at a higher resolution.

Stability of recovering from noisy data. In this simulation, we show the recovery performance with respect to noisy data. Again, we fixed $n_1 = 100, n_2 = 150, n_3 = 200$ and $r_1 = r_2 = r_3 = r$ and tested against different choices of (r, m). For each choice of (r, m), we sampled the ground truth $\mathbf{A}, \mathbf{B}, \mathbf{C}$ using the same method as in the previous simulation. We generated Ω uniformly at random. For each entry $(i, j, k) \in \Omega$, we simulated the noisy observation $\hat{T}_{ijk} = T_{ijk} + \epsilon_{ijk}$, where ϵ_{ijk} is a zero-mean Gaussian random variable with variance σ_n^2. Then, we revealed

$\{\hat{T}_{ijk}\}_{(ijk)\in\Omega}$ to the noisy recovery algorithm and collect the recovered matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. The error of recovery result is measured by $(\|\mathbf{X}-\mathbf{A}\|_F+\|\mathbf{Y}-\mathbf{B}\|_F+\|\mathbf{Z}-\mathbf{C}\|_F)/(\|\mathbf{A}\|_F+\|\mathbf{B}\|_F+\|\mathbf{C}\|_F)$. We tested the algorithm with a range of noise levels and for each different configuration of (r, m, σ_n^2), we repeated the experiments for 10 times and recorded the mean and standard deviation of the relative error.

noise level	relative error
0.1	0.1020 ± 0.0005
0.2	0.1972 ± 0.0007
0.3	0.2877 ± 0.0011
0.4	0.3720 ± 0.0015
0.5	0.4524 ± 0.0015

(a) Fix $r = 20$, $m = 5d$ and noise level varies.

observations m	relative error
$m = 3d$	0.1445 ± 0.0008
$m = 4d$	0.1153 ± 0.0006
$m = 5d$	0.1015 ± 0.0004
$m = 6d$	0.0940 ± 0.0007
$m = 7d$	0.0920 ± 0.0011

(b) Fix $r = 20$, 0.1 noise level and m varies.

rank r	relative error
10	0.1134 ± 0.0006
20	0.1018 ± 0.0007
30	0.0973 ± 0.0037
40	0.1032 ± 0.0212
50	0.1520 ± 0.0344

(c) Fix $m = 5d$, 0.1 noise level and r varies.

Table 5.1: Simulation results of noisy data.

We present the result of the experiments in Table 5.1. From the results in Table 5.1(a), we can see that the error in the solution is proportional to the noise level. Table 5.1(b) indicates that the recovery is not reliable when we have too few observations, while the performance of the algorithm is much more stable for a sufficient number of observations around four times of the degree of freedom. Table 5.1(c) shows that the recovery error is not affected much by the rank, as the number of observations scales with the degree of freedom in our setting.

Temporal collaborative filtering. In order to demonstrate the performance of pairwise interaction tensor on real world applications, we conducted experiments on the Movielens dataset. The MovieLens dataset contains 1,000,209 ratings from 6,040 users and 3,706 movies from April, 2000 and February, 2003. Each rating from Movielens dataset is accompanied with time information provided in seconds. We transformed each timestamp into its corresponding calendar month. We randomly select 10% ratings as test set and use the rest of the ratings as training set. In the end, we obtained a tensor $\mathcal{T}$ of size $6040 \times 3706 \times 36$, in which the axes corresponded to user, movie and timestamp respectively, with 0.104% observed entries as the training set. We applied the noisy recovery algorithm on the training set. Following previous studies which applies SVT algorithm on movie recommendation datasets [76], we used a pre-specified truncation level r for computing SVD in each iteration, i.e., we only kept top r singular vectors. Therefore, the rank of recovered matrices are at most r.

We evaluated the prediction performance in terms of root mean squared error (RMSE). We compared our algorithm with noisy matrix completion method using standard SVT optimization algorithm [30, 31] to the same dataset while ignore the time information. Here we can regard the noisy matrix completion algorithm as a special case of the recover a pairwise interaction tensor of size $6040 \times 3706 \times 1$, i.e., the time information is ignored. We also noted that the training tensor had more than one million observed entries and 80 millions total entries. This scale made a number of tensor recovery algorithms, for example Tucker decomposition and PARAFAC [85], impractical to apply on the dataset. In contrast, our recovery algorithm took 2430 seconds to finish on a standard workstation for truncation level $r = 100$.

The experimental result is shown in Figure 5.2. The empirical result of Figure 5.2(a) suggests that, by incorporating the temporal information, pairwise interaction tensor recovery algorithm consistently outperformed the matrix completion method. Interestingly, we can see that, for most parameter settings in Figure 5.2(b), our algorithm recovered a rank 2 matrix $\mathbf{Y}$ representing the change of movie popularity over time and a rank 15 matrix $\mathbf{Z}$ that encodes the change of user interests over time. The reason of the improvement on the prediction performance may be that the recovered matrix $\mathbf{Y}$ and $\mathbf{Z}$ provided meaningful signal. Finally, we note that our algorithm achieves a RMSE of 0.858 when the truncation level is set to 50, which slightly outperforms the RMSE=0.861 (quote from Figure 7 of the paper) result of 30-dimensional Bayesian Probabilistic Tensor Factorization (BPTF) on the same dataset, where the authors predict the ratings by factorizing a $6040 \times 3706 \times 36$ tensor using BPTF method [152]. We may attribute the performance gain to the modeling flexibility of pairwise interaction tensor and the learning guarantees of our algorithm.

5.5 Conclusion

In this chapter, we proved rigorous guarantees for convex programs for recovery of pairwise interaction tensors with missing entries, both in the absence and in the presence of noise. We designed a scalable optimization algorithm for solving the convex programs. We supplemented our theoretical results with simulation experiments and a real-world application to movie recommendation. In the noiseless case, simulations showed that the exact recovery almost always succeeded if the number of observations is a constant time of the degree of freedom, which agrees asymptotically with the theoretical result. In the noisy case, the simulation results confirmed that the stable recovery algorithm is able to reliably recover pairwise interaction tensor from noisy observations. Our results on the temporal movie recommendation application demonstrated that, by incorporating the temporal information, our algorithm outperforms conventional matrix completion and achieves state-of-the-art results.

5.6 Proof of Theorem 5.1

Sampling model. Recall that Theorem 5.1 assumed that Ω is sampled uniformly at random from the collection of all set of size m. This uniform sampling model turns out to be awkward to deal with. Following the strategy of [68, 119], we use the sampling with replacement model on Ω as a proxy for uniform sampling. This differs from the earlier approach by [32] where the authors used a Bernoulli sampling model as a proxy for uniform sampling model. The sampling with replacement model has enabled a significant simplification on the proof and therefore we shall follow this model in the rest of our proof. Specifically, we consider the case where the index of each observation is sampled independently and uniformly from the set $[n_1] \times [n_2] \times [n_3]$. Note that, in expectation, the sampling with replacement model is the same with uniform sampling model. It may appear to be troublesome since the sampling with replacement model can lead to duplicated entires. However, the following lemma allows us to bound the probability of failure when sampling with replacement by the likelihood of error under uniform sampling model.

Lemma 5.1. *([119, Proposition 3.1]) The probability that the recovery algorithm Eq.* (5.3) *fails when Ω is sampled uniformly* *from the collection of sets of size m is no larger than the probability that the algorithm fails when each index of Ω is sampled independently and uniformly.*

Proof. The proof is similar to [36, Section ii.C] and [119, Proposition 3.1]. Let Ω' be a collection of indices sampled independent and uniformly from the set $[n_1] \times [n_2] \times [n_3]$. Also denote Ω_k as a set of entries of size k sampled uniformly at random from all sets of entries of size k. Then, we have

$$\begin{aligned}
\Pr(\text{Failure}(\Omega')) &= \sum_{k=0}^{m} \Pr(\text{Failure}(\Omega')||\Omega'| = k) \Pr(|\Omega'| = k) \\
&= \sum_{k=0}^{m} \Pr(\text{Failure}(\Omega_k)) \Pr(|\Omega'| = k) \\
&\geq \Pr(\text{Failure}(\Omega_m)) \sum_{k=0}^{m} \Pr(|\Omega'| = k) \\
&= \Pr(\text{Failure}(\Omega_m)).
\end{aligned}$$

□

Therefore, the probability of failure when sampling with replacement is larger than that under uniform sampling model. Hence, we only need to upper bound the failure probability under sampling with replacement model to prove Theorem 5.1. In the rest of this chapter, we will consider solely sampling with replacement model.

Preliminaries. In order to present the proof, we require several additional notations. We shall slightly abuse the notation and denote $\mathbf{e}_k$ be the kth standard basis vector, equal to 1 in kth entry and 0 everywhere else. Denote $\delta_{ij} = \mathbf{e}_i\mathbf{e}_j^T$ be the matrix which equals to 1 in entry (i, j) and 0 in other entries. The dimension of $\mathbf{e}_k$ and δ_{ij} shall be clear from context.

Let $\Omega = \{(a_i, b_i, c_i)\}_{i\in[m]}$, where each (a_i, b_i, c_i) is sampled independently and uniformly at random from $[n_1] \times [n_2] \times [n_3]$.

We define the operator $\mathcal{R}_\Omega : \mathbb{R}^{n_1\times n_2} \otimes \mathbb{R}^{n_2\times n_3} \otimes \mathbb{R}^{n_3\times n_1} \to \mathbb{R}^m$ to be

$$\mathcal{R}_\Omega(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \sum_{i=1}^m \frac{1}{\sqrt{n_3}} \langle \mathbf{X}, \delta_{a_i b_i}\rangle + \frac{1}{\sqrt{n_1}} \langle \mathbf{Y}, \delta_{b_i c_i}\rangle + \frac{1}{\sqrt{n_2}} \langle \mathbf{Z}, \delta_{c_i a_i}\rangle . \tag{5.11}$$

Then, the original convex program Eq. (5.3) can be reformulated as

$$\begin{aligned} \underset{\mathbf{X}\in S_A, \mathbf{Y}\in S_B, \mathbf{Z}\in S_C}{\text{minimize}} \quad & \|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_* \\ \text{subject to} \quad & \mathcal{R}_\Omega(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \mathbf{t}, \end{aligned} \tag{5.12}$$

where $t_i = T_{a_i b_i c_i}$ is the ith observation of $\mathcal{T}$. Note that the scaling coefficients on $\mathbf{X},\mathbf{Y},\mathbf{Z}$ have been incorporated into $\mathcal{R}_\Omega$.

In order to further simplify the notations, we consider the following block diagonal matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{X} & & \\ & \mathbf{Y} & \\ & & \mathbf{Z} \end{bmatrix},$$

or more compactly $\mathbf{M} = \operatorname{diag}(\mathbf{X},\mathbf{Y},\mathbf{Z})$. It is clear that $\|\mathbf{M}\|_* = \|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_*$. Now, denote $\delta_{ab}^{(A)} \triangleq \operatorname{diag}(\delta_{ab}, \mathbf{0}_{n_2\times n_3}, \mathbf{0}_{n_3\times n_1})$ where $\delta_{ab}^{(A)}$ is a $n_1 \times n_2$ matrix and $\mathbf{0}_{n_2\times n_3}$ and $\mathbf{0}_{n_3\times n_1}$ are zero matrices of size $n_2 \times n_3$ and $n_3 \times n_1$, respectively. Similarly, we define define $\delta_{bc}^{(B)} \triangleq \operatorname{diag}(\mathbf{0}_{n_1\times n_2}, \delta_{bc}, \mathbf{0}_{n_3\times n_1})$ and $\delta_{ca}^{(C)} \triangleq \operatorname{diag}(\mathbf{0}_{n_1\times n_2}, \mathbf{0}_{n_2\times n_3}, \delta_{ca})$. Now, we have

$$\mathbf{M} = \sum_{ab} \left\langle \mathbf{X}, \delta_{ab}^{(A)} \right\rangle + \sum_{bc} \left\langle \mathbf{Y}, \delta_{bc}^{(B)} \right\rangle + \sum_{ca} \left\langle \mathbf{Z}, \delta_{ca}^{(C)} \right\rangle .$$

Then, we may equivalently define $\mathcal{R}_\Omega$ by

$$\mathcal{R}_\Omega(\mathbf{M}) = \sum_{i=1}^m \frac{1}{\sqrt{n_3}} \left\langle \mathbf{M}, \delta_{a_i b_i}^{(A)} \right\rangle + \frac{1}{\sqrt{n_1}} \left\langle \mathbf{M}, \delta_{b_i c_i}^{(B)} \right\rangle + \frac{1}{\sqrt{n_2}} \left\langle \mathbf{M}, \delta_{c_i a_i}^{(C)} \right\rangle$$

$$
\begin{aligned}
&= \sum_{i=1}^{m} \left\langle \mathbf{M}, \frac{1}{\sqrt{n_3}}\delta^{(A)}_{a_i b_i} + \frac{1}{\sqrt{n_1}}\delta^{(B)}_{b_i c_i} + \frac{1}{\sqrt{n_2}}\delta^{(C)}_{c_i a_i} \right\rangle \\
&\triangleq \sum_{i=1}^{m} \left\langle \mathbf{M}, \sigma_{a_i b_i c_i} \right\rangle, \qquad (5.13)
\end{aligned}
$$

where in the last equation, we have defined

$$
\sigma_{a_i b_i c_i} \triangleq \frac{1}{\sqrt{n_3}}\delta^{(A)}_{a_i b_i} + \frac{1}{\sqrt{n_1}}\delta^{(B)}_{b_i c_i} + \frac{1}{\sqrt{n_2}}\delta^{(C)}_{c_i a_i}. \qquad (5.14)
$$

Note that we have $\mathcal{R}_\Omega(\mathbf{M}) = \mathcal{R}_\Omega(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. Therefore, we can further rewrite the convex program as

$$
\begin{aligned}
\underset{\mathbf{M}\in S}{\text{minimize}} \quad & \|\mathbf{M}\|_* \qquad (5.15) \\
\text{subject to} \quad & \mathcal{R}_\Omega(\mathbf{M}) = \mathbf{t},
\end{aligned}
$$

where we have define the linear subspace S as the product space of S_A, S_B and S_C, namely,

$$
S = \{\mathrm{diag}(\mathbf{A}, \mathbf{B}, \mathbf{C}) : \mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C\}.
$$

Hence, $\mathbf{M} \in S$ if and only if $\mathbf{M}$ is a block diagonal matrix $\mathrm{diag}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and $\mathbf{X} \in S_A$, $\mathbf{Y} \in S_B$ and $\mathbf{Z} \in S_C$. For convenience, we also define the orthogonal complement $S^\perp$ by

$$
S^\perp = \{\mathrm{diag}(\mathbf{A}, \mathbf{B}, \mathbf{C}) : \mathbf{A} \in S_A^\perp, \mathbf{B} \in S_B^\perp, \mathbf{C} \in S_C^\perp\}.
$$

Indeed, the convex program Eq. (5.15), despite the constraint $\mathbf{M} \in S$, seems to be very similar to the standard nuclear norm heuristic to matrix completion problem. However, we found the major challenge here is that the observation operator $\mathcal{R}_\Omega$ is non-orthogonal. Previously, Gross et al. [68] showed that the nuclear norm heuristic leads to exact recovery when the observation operator is orthogonal. The orthogonality of observation operator is critical to their argument and therefore their proof cannot be directly applied to our problem. In this work, we extend their technique to deal with the non-orthogonal operator $\mathcal{R}_\Omega$. It turns out that the constraint $\mathbf{M} \in S$ plays a vital role in the argument which is unknown in previous work of matrix completion.

In the rest of chapter, we shall consider the following singular value decompositions of $\mathbf{A}, \mathbf{B}, \mathbf{C}$

$$\mathbf{A} = \mathbf{U}_A \Sigma_A \mathbf{V}_A^T, \quad \mathbf{B} = \mathbf{U}_B \Sigma_B \mathbf{V}_B^T, \quad \mathbf{C} = \mathbf{U}_C \Sigma_C \mathbf{V}_C^T.$$

Recall that we have defined $S_B = \{\mathbf{M} \in \mathbb{R}^{n_2 \times n_3} : \mathbf{1}^T \mathbf{M} = \mathbf{0}^T\}$, $S_C = \{\mathbf{M} \in \mathbb{R}^{n_3 \times n_1} : \mathbf{1}^T \mathbf{M} = \mathbf{0}^T\}$ and $S_A = \{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2} : \mathbf{1}^T \mathbf{M} = \left(\frac{1}{n_2} \mathbf{1}^T \mathbf{M} \mathbf{1}\right) \mathbf{1}^T\}$. Now, we introduce the orthogonal decompositions of $S_A = T_A \oplus T_A^\perp$, $S_B = T_B \oplus T_B^\perp$ and $S_C = T_C \oplus T_C^\perp$, where T_A is the linear space $T_A = \{\mathbf{U}_A \mathbf{Y}^T + \mathbf{X} \mathbf{V}_A^T : \forall \mathbf{X}, \mathbf{Y}\} \cap S_A$ and $T_A^\perp$ is the orthogonal complement (and respectively $T_B, T_C, T_B^\perp, T_C^\perp$) are defined similarly. Analogous to the definition of S, we define subspace T as

$$T = \{\mathrm{diag}(\mathbf{A}, \mathbf{B}, \mathbf{C}) : \mathbf{A} \in T_A, \mathbf{B} \in T_B, \mathbf{C} \in T_C\}.$$

We also denote the orthogonal complement of T as $T^\perp$ which is defined in a similar way.

Further, the orthogonal projection operator onto $\mathcal{T}_A$ is given by

$$\mathcal{P}_{T_A}(\mathbf{Z}) = \mathbf{P}_{U_A} \mathcal{P}_{S_A}(\mathbf{Z}) + \mathcal{P}_{S_A}(\mathbf{Z}) \mathbf{P}_{V_A} - \mathbf{P}_{U_A} \mathcal{P}_{S_A}(\mathbf{Z}) \mathbf{P}_{V_A},$$

where $\mathbf{P}_{U_A}$, $\mathbf{P}_{V_A}$ are the orthogonal projections onto U_A and V_A respectively and $\mathcal{P}_{S_A}$ is the orthogonal projection onto S_A. By simple calculation, we can derive

$$\mathcal{P}_{S_A}(\mathbf{A}) = \mathbf{A} - \frac{1}{n_1} \mathbf{1}^T \mathbf{1} \mathbf{A} + \frac{1}{n_1 n_2} (\mathbf{1}^T \mathbf{A} \mathbf{1}) \mathbf{1}^T \mathbf{1}.$$

Similarly, we can derive the orthogonal projection operator $\mathcal{P}_{T_B}$ (respectively $\mathcal{P}_{T_C}$) onto $\mathcal{T}_B$ (respectively $\mathcal{T}_C$) as $\mathcal{P}_{T_B}(\mathbf{Z}) = \mathbf{P}_{U_B} \mathcal{P}_{S_B}(\mathbf{Z}) + \mathcal{P}_{S_B}(\mathbf{Z}) \mathbf{P}_{V_B} - \mathbf{P}_{U_B} \mathcal{P}_{S_B}(\mathbf{Z}) \mathbf{P}_{V_B}$ and $\mathcal{P}_{T_C}(\mathbf{Z}) = \mathbf{P}_{U_C} \mathcal{P}_{S_C}(\mathbf{Z}) + \mathcal{P}_{S_C}(\mathbf{Z}) \mathbf{P}_{V_C} -$

$\mathbf{P}_{U_C}\mathcal{P}_{S_C}(\mathbf{Z})\mathbf{P}_{V_C}$, where $\mathcal{P}_{S_B}(\mathbf{B}) = \mathbf{B} - \frac{1}{n_2}\mathbf{1}^T\mathbf{1}\mathbf{B}$ and $\mathcal{P}_{S_C}(\mathbf{C}) = \mathbf{C} - \frac{1}{n_3}\mathbf{1}^T\mathbf{1}\mathbf{C}$.

In addition, we also consider the orthogonal decomposition $\mathbb{R}^{n_1\times n_2} = S_A \oplus S_A^\perp$ (respectively S_B,$S_B^\perp$, S_C, $S_C^\perp$). The orthogonal projection operator $\mathcal{P}_{S_A^\perp}$, $\mathcal{P}_{S_B^\perp}$ and $\mathcal{P}_{S_C^\perp}$ are given by

$$\mathcal{P}_{S_A^\perp}(\mathbf{A}) = (\mathcal{I} - \mathcal{P}_{S_A})(\mathbf{A}) = \frac{1}{n_1}\mathbf{1}^T\mathbf{1}\mathbf{A} - \frac{1}{n_1 n_2}(\mathbf{1}^T\mathbf{A}\mathbf{1})\mathbf{1}^T\mathbf{1},$$

and

$$\mathcal{P}_{S_B^\perp}(\mathbf{B}) = \frac{1}{n_2}\mathbf{1}^T\mathbf{1}\mathbf{B}, \quad \mathcal{P}_{S_C^\perp}(\mathbf{C}) = \frac{1}{n_3}\mathbf{1}^T\mathbf{1}\mathbf{C}.$$

Moreover, we can derive the orthogonal projection operators $\mathcal{P}_S$ as $\mathcal{P}_S(\mathrm{diag}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})) = \mathrm{diag}(\mathcal{P}_{S_A}(\mathbf{X}), \mathcal{P}_{S_B}(\mathbf{Y}), \mathcal{P}_{S_C}(\mathbf{Z}))$. The orthogonal projection operators $\mathcal{P}_T, \mathcal{P}_{S^\perp}$ and $\mathcal{P}_{T^\perp}$ can also be derived similarly.

To proceed, we shall need one additional tool, the non-commutative Bernstein inequality.

Theorem 5.3. *(Non-commutative Bernstein inequality [119, Theorem 3.2]) Let $\mathbf{X}_1, \ldots, \mathbf{X}_m$ be independent zero mean random matrices of dimension $d_1 \times d_2$. Suppose $\rho_k^2 = \max\{\left\|\mathbf{E}[\mathbf{X}_k\mathbf{X}_k^T]\right\|, \left\|\mathbf{E}[\mathbf{X}_k^T\mathbf{X}_k]\right\|\}$ and $\|\mathbf{X}_k\| \le M$ almost surely for every k. Then, for any $\tau > 0$,*

$$\Pr\left[\left\|\sum_{k=1}^m \mathbf{X}_k\right\| > \tau\right] \le (d_1 + d_2)\exp\left(\frac{-\tau^2/2}{\sum_{k=1}^m \rho_k^2 + M\tau/3}\right).$$

We omit the proof of the non-commutative Bernstein inequality. For details, readers may refer to [119, Appendix A] and [4]. Furthermore, the righthand side is always less than $(d_1 + d_2)\exp(-\frac{3}{8}\tau^2/(\sum_{k=1}^m \rho_k^2))$ when $\tau \le \frac{1}{M}\sum_{k=1}^m \rho_k^2$. In our proof, we will solely rely on the condensed version of non-commutative Bernstein inequality.

We are now ready to state the proof of Theorem 5.1. First, in the following theorem, we show that if there exists a "dual certificate", the solution to convex program Eq. (5.3) is unique and exactly recovers the pairwise interaction tensor.

Theorem 5.4. *Let* $r = \max\{r_1, r_2, r_3\}$*. Let* $\mathbf{W} = \operatorname{diag}(\mathbf{U}_A\mathbf{V}_A^T, \mathbf{U}_B\mathbf{V}_B^T, \mathbf{U}_C\mathbf{V}_C^T)$ *be a block diagonal matrix. Suppose that there exists a "dual certificate"* $\mathbf{F} \in \operatorname{range}(\mathcal{R}_\Omega^*)$ *such that*

$$\|\mathcal{P}_T(\mathbf{F}) - \mathbf{W}\|_F \leq \sqrt{\frac{r}{2n_3}}, \quad \|\mathcal{P}_{T^\perp}(\mathbf{F})\| < \frac{1}{2}$$

And also suppose that

$$\frac{1}{2}\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* > \sqrt{\frac{r}{2n_3}}\|\mathcal{P}_T(\mathbf{E})\|_F$$

holds for any $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$*. Then, the* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *is the unique minimizing solution of Eq.* (5.3)*.*

Proof. Let $\mathbf{M} = \operatorname{diag}\left(\sqrt{n_3}\mathbf{A}, \sqrt{n_1}\mathbf{B}, \sqrt{n_2}\mathbf{C}\right)$ be a block diagonal matrix. By the definition of nuclear norm, we have $\|\mathbf{M}\|_* = \sqrt{n_3}\|\mathbf{A}\|_* + \sqrt{n_1}\|\mathbf{B}\|_* + \sqrt{n_2}\|\mathbf{C}\|_*$. Now, consider for any block diagonal matrix $\mathbf{E} = \operatorname{diag}(\mathbf{E}_A, \mathbf{E}_B, \mathbf{E}_C)$ such that $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$. Pick $\mathbf{U}_{A^\perp}$ and $\mathbf{V}_{A^\perp}$ such that $[\mathbf{U}_A, \mathbf{U}_{A^\perp}]$ and $[\mathbf{V}_A, \mathbf{V}_{A^\perp}]$ are unitary matrices and $\left\langle \mathbf{U}_{A^\perp}\mathbf{V}_{A^\perp}^T, \mathcal{P}_{T_A^\perp}(\mathbf{E}_A)\right\rangle = \left\|\mathcal{P}_{T_A^\perp}(\mathbf{E})\right\|_*$. Also pick $\mathbf{U}_{B^\perp}, \mathbf{V}_{B^\perp}, \mathbf{U}_{C^\perp}, \mathbf{V}_{C^\perp}$ similarly. Let $\mathbf{W}_\perp = \operatorname{diag}(\mathbf{U}_{A^\perp}\mathbf{V}_{A^\perp}^T, \mathbf{U}_{B^\perp}\mathbf{V}_{B^\perp}^T, \mathbf{U}_{C^\perp}\mathbf{V}_{C^\perp}^T)$. We have $\mathbf{W}_\perp \in T^\perp$ and $\langle \mathbf{W}_\perp, \mathcal{P}_{T^\perp}(\mathbf{E})\rangle = \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_*$. Also note the fact that $\langle \mathbf{F}, \mathbf{E}\rangle = 0$ since $\mathbf{F} \in \operatorname{range}(\mathcal{R}_\Omega^T)$ and $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$. Then it follows that,

$$\begin{aligned}
&\|\mathbf{M} + \mathbf{E}\|_* \\
\geq &\langle \mathbf{W} + \mathbf{W}_\perp, \mathbf{M} + \mathbf{E}\rangle \\
= &\|\mathbf{M}\|_* + \langle \mathbf{W} + \mathbf{W}_\perp, \mathbf{E}\rangle \\
= &\|\mathbf{M}\|_* + \langle \mathbf{W} + \mathbf{W}_\perp - \mathbf{F}, \mathbf{E}\rangle \\
= &\|\mathbf{M}\|_* + \langle \mathbf{W} + \mathbf{W}_\perp - \mathbf{F}, \mathcal{P}_T(\mathbf{E}) + \mathcal{P}_{T^\perp}(\mathbf{E}) + \mathcal{P}_{S^\perp}(\mathbf{E})\rangle \\
= &\|\mathbf{M}\|_* + \langle \mathbf{W} - \mathcal{P}_T(\mathbf{F}), \mathcal{P}_T(\mathbf{E})\rangle + \langle \mathbf{W}_\perp - \mathcal{P}_{T^\perp}(\mathbf{F}), \mathcal{P}_{T^\perp}(\mathbf{E})\rangle \\
= &\|\mathbf{M}\|_* - \langle \mathcal{P}_T(\mathbf{F}) - \mathbf{W}, \mathcal{P}_T(\mathbf{E})\rangle + \langle \mathbf{W}_\perp, \mathcal{P}_{T^\perp}(\mathbf{E})\rangle \\
&\quad - \langle \mathcal{P}_{T^\perp}(\mathbf{F}), \mathcal{P}_{T^\perp}(\mathbf{E})\rangle
\end{aligned}$$

$$\begin{aligned}
&\geq \|\mathbf{M}\|_* - \|\mathcal{P}_T(\mathbf{F}) - \mathbf{W}\|_F \|\mathcal{P}_T(\mathbf{E})\|_F + \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* \\
&\qquad - \|\mathcal{P}_{T^\perp}(\mathbf{F})\| \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* \\
&> \|\mathbf{M}\|_* - \sqrt{\frac{r}{2n_3}} \|\mathcal{P}_T(\mathbf{E})\|_F + \frac{1}{2} \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* \\
&\geq \|\mathbf{M}\|_* = \sqrt{n_3} \|\mathbf{A}\|_* + \sqrt{n_1} \|\mathbf{B}\|_* + \sqrt{n_2} \|\mathbf{C}\|_* ,
\end{aligned}$$

where the first inequality follows from the variational characterization of nuclear norm $\|\mathbf{M} + \mathbf{E}\|_* = \sup_{\|\mathbf{Q}\|=1} \langle \mathbf{Q}, \mathbf{M} + \mathbf{E} \rangle$. We have also used the fact that $\mathbf{E} \in S$ and therefore $\mathcal{P}_{S^\perp}(\mathbf{E}) = \mathbf{0}$. Therefore, if there exists any $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ obeying $\mathcal{R}_\Omega(\mathrm{diag}(\sqrt{n_3}(\mathbf{X} - \mathbf{A}), \sqrt{n_1}(\mathbf{Y} - \mathbf{B}), \sqrt{n_2}(\mathbf{Z} - \mathbf{C})) = \mathbf{0}$, or equivalently $\frac{1}{\sqrt{n_3}}\sqrt{n_3}(X_{ij} - A_{ij}) + \frac{1}{\sqrt{n_1}}\sqrt{n_1}(Y_{jk} - B_{jk}) + \frac{1}{\sqrt{n_2}}\sqrt{n_2}(Z_{ki} - C_{ki}) = 0$ for all $(i, j, k) \in \Omega$, we would have $\|\mathbf{X}\|_* + \|\mathbf{Y}\|_* + \|\mathbf{Z}\|_* > \|\mathbf{A}\|_* + \|\mathbf{B}\|_* + \|\mathbf{C}\|_*$. In other words, if $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ statisifes $X_{ij} + Y_{jk} + Z_{ki} = A_{ij} + B_{jk} + C_{ki}$ for any $(i, j, k) \in \Omega$, the weighted sum of the nuclear norm of $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ would be strictly larger than that of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$. Therefore, $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ is the unique minimizer of program Eq. (5.3). □

Therefore, we remain to show that such a dual certificate $\mathbf{F}$ exists with high probability. The proof relies on a series of applications of noncommutative Bernstein inequality and the clever golfing scheme proposed by [68]. We begin with an elementary bound on $\|\mathcal{P}_T(\sigma_{abc})\|_F$.

Proposition 5.2. *Suppose that $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are (μ_0, μ_1)-incoherent. Then, for any $(a, b, c) \in [n_1] \times [n_2] \times [n_3]$, the length of orthogonal projection of σ_{abc} onto space T is bounded by*

$$\|\mathcal{P}_T(\sigma_{abc})\|_F^2 \leq \frac{28\mu_0 r}{n_1 n_2},$$

where $\sigma_{abc} \triangleq \frac{1}{\sqrt{n_3}}\delta_{ab}^{(A)} + \frac{1}{\sqrt{n_1}}\delta_{bc}^{(B)} + \frac{1}{\sqrt{n_2}}\delta_{ca}^{(C)}$.

Proof. By definition, we have $\left\|\mathcal{P}_T\left(\sigma_{abc}\right)\right\|_F^2 = \frac{1}{n_3}\left\|\mathcal{P}_{T_A}(\delta_{ab})\right\|_F^2 + \frac{1}{n_1}\left\|\mathcal{P}_{T_B}(\delta_{bc})\right\|_F^2 + \frac{1}{n_2}\left\|\mathcal{P}_{T_C}(\delta_{ca})\right\|_F^2$. Therefore, it suffices to bound these terms individually.

We first deal with $\left\|\mathcal{P}_{T_B}(\delta_{bc})\right\|_F$. It can be decomposed as

$$\begin{aligned}
&\left\|\mathcal{P}_{T_B}(\delta_{bc})\right\|_F^2 \\
=&\langle \mathcal{P}_{T_B}(\delta_{bc}), \mathcal{P}_{T_B}(\delta_{bc})\rangle \\
=&\langle \mathcal{P}_{T_B}(\delta_{bc}), \delta_{bc}\rangle \\
=&\left\|\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc})\right\|_F^2 + \left\|\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2 - \left\|\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2 \\
\leq&\left\|\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc})\right\|_F^2 + \left\|\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2 .
\end{aligned}$$

Now it suffices to bound both terms $\left\|\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc})\right\|_F^2$ and $\left\|\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2$. We have

$$\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc}) = \mathbf{P}_{U_B}\delta_{bc} - \frac{1}{n_3}\mathbf{P}_{U_B}\mathbf{1}\mathbf{1}^T\delta_{bc} = \mathbf{P}_{U_B}\delta_{bc},$$

where the second equality holds since $\mathbf{B} \in S_B$ and therefore $\mathbf{P}_{U_B}\mathbf{1} = \mathbf{0}$. Combining with the incoherence property **A0**, we have

$$\left\|\mathbf{P}_{U_B}\mathcal{P}_{S_B}(\delta_{bc})\right\|_F^2 = \left\|\mathbf{P}_{U_B}\delta_{bc}\right\|_F^2 = \left\|\mathbf{P}_{U_B}\mathbf{e}_b\right\|_F^2 \leq \frac{\mu_0 r}{n_2}.$$

Next, we need to bound $\left\|\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2$. We have

$$\begin{aligned}
\left\|\mathcal{P}_{S_B}(\delta_{bc})\mathbf{P}_{V_B}\right\|_F^2 &= \left\|\delta_{bc}\mathbf{P}_{V_B} - \frac{1}{n_2}\mathbf{1}\mathbf{1}^T\delta_{bc}\mathbf{P}_{V_B}\right\|_F^2 \\
&\leq 2\left\|\delta_{bc}\mathbf{P}_{V_B}\right\|_F^2 + \frac{2}{n_2^2}\left\|\mathbf{1}\mathbf{1}^T\delta_{bc}\mathbf{P}_{V_B}\right\|_F^2 \\
&= 2\left\|\mathbf{P}_{V_B}\mathbf{e}_c\right\|_F^2 + \frac{2}{n_2^2}\left\|\mathbf{1}\mathbf{e}_c^T\mathbf{P}_{V_B}\right\|_F^2 \\
&= 2\left\|\mathbf{P}_{V_B}\mathbf{e}_c\right\|_F^2 + \frac{2}{n_2^2}\left\|\sum_{b'}\mathbf{e}_{b'}\mathbf{e}_c^T\mathbf{P}_{V_B}\right\|_F^2 \\
&= 2\left\|\mathbf{P}_{V_B}\mathbf{e}_c\right\|_F^2 + \frac{2}{n_2^2}\left\|n_2\mathbf{P}_{V_B}\mathbf{e}_c\right\|_F^2
\end{aligned}$$

$$
\begin{aligned}
&= 4\left\|\mathbf{P}_{V_B}\mathbf{e}_c\right\|_F^2\\
&\le \frac{4\mu_0 r}{n_3},
\end{aligned}
$$

where the first inequality is the Cauchy-Schwartz inequality and the final inequality is due to incoherence property **A0**. Therefore,

$$
\left\|\mathcal{P}_{T_B}(\delta_{bc})\right\|_F^2 \le \frac{\mu_0 r}{n_2} + \frac{4\mu_0 r}{n_3} \le \frac{5\mu_0 r}{n_2}.
$$

In addition, we can bound $\left\|\mathcal{P}_{T_C}(\delta_{ca})\right\|_F$ using the same method.

It remains to bound $\left\|\mathcal{P}_{T_A}(\delta_{ab})\right\|_F^2$ which is no greater than $\left\|\mathbf{P}_{U_A}\mathcal{P}_{S_A}(\delta_{ab})\right\|_F^2 + \left\|\mathcal{P}_{S_A}(\delta_{ab})\mathbf{P}_{V_A}\right\|_F^2$ following a similar analysis. Again, we start with bounding $\left\|\mathbf{P}_{U_A}\mathcal{P}_{S_A}(\delta_{ab})\right\|_F^2$. We have

$$
\begin{aligned}
&\left\|\mathbf{P}_{U_A}\mathcal{P}_{S_A}(\delta_{ab})\right\|_F^2\\
=&\left\|\mathbf{P}_{U_A}\delta_{ab} - \frac{1}{n_1}\mathbf{P}_{U_A}\mathbf{1}\mathbf{e}_b^T + \frac{1}{n_1 n_2}\mathbf{P}_{U_A}\mathbf{1}\mathbf{1}^T\right\|_F^2\\
\le&3\left\|\mathbf{P}_{U_A}\delta_{ab}\right\|_F^2 + \frac{3}{n_1^2}\left\|\mathbf{P}_{U_A}\mathbf{1}\mathbf{e}_b^T\right\|_F^2 + \frac{3}{n_1^2 n_2^2}\left\|\mathbf{P}_{U_A}\mathbf{1}\mathbf{1}^T\right\|_F^2\\
=&3\left\|\mathbf{P}_{U_A}\mathbf{e}_a\right\|_F^2 + \frac{3}{n_1^2}\left\|\sum_{a'}\mathbf{P}_{U_A}\mathbf{e}_{a'}\right\|_F^2 + \frac{3}{n_1^2 n_2^2}\left\|\sum_{a'b'}\mathbf{P}_{U_A}\mathbf{e}_{a'}\right\|_F^2\\
\le&3\left\|\mathbf{P}_{U_A}\mathbf{e}_a\right\|_F^2 + \frac{3}{n_1}\sum_{a'}\left\|\mathbf{P}_{U_A}\mathbf{e}_{a'}\right\|_F^2 + \frac{3}{n_1 n_2}\sum_{a'b'}\left\|\mathbf{P}_{U_A}\mathbf{e}_{a'}\right\|_F^2\\
\le&\frac{9\mu_0 r}{n_1},
\end{aligned}
$$

where we have repeatedly applied Cauchy-Schwartz inequality and assumption A1. We can also bound $\left\|\mathbf{P}_{S_A}(\delta_{ab})\mathbf{P}_{V_A}\right\|_F^2$ using the same method as

$$
\left\|\mathbf{P}_{S_A}(\delta_{ab})\mathbf{P}_{V_A}\right\|_F^2 \le \frac{9\mu_0 r}{n_2}.
$$

Therefore, we have

$$
\left\|\mathcal{P}_{T_A}(\delta_{ab})\right\|_F^2 \le \frac{9\mu_0 r}{n_1} + \frac{9\mu_0 r}{n_2} \le \frac{18\mu_0 r}{n_1}.
$$

Finally, combining the above inequalities, we have

$$\begin{aligned}
&\left\|\mathcal{P}_T(\sigma_{abc})\right\|_F^2\\
=&\frac{1}{n_3}\left\|\mathcal{P}_{T_A}(\delta_{ab})\right\|_F^2+\frac{1}{n_1}\left\|\mathcal{P}_{T_B}(\delta_{bc})\right\|_F^2+\frac{1}{n_2}\left\|\mathcal{P}_{T_C}(\delta_{ca})\right\|_F^2\\
\leq&\frac{26\mu_0 r}{n_1 n_2}\\
\leq&\frac{18\mu_0 r}{n_1 n_3}+\frac{5\mu_0 r}{n_1 n_2}+\frac{5\mu_0 r}{n_1 n_2}\\
\leq&\frac{28\mu_0 r}{n_1 n_2}.
\end{aligned}$$

□

The next proposition shows that, in expectation, $\frac{n_1 n_2 n_3}{m}\mathcal{R}_\Omega^*\mathcal{R}_\Omega$ is an isometric operator on S. Therefore, the observation operator $\mathcal{R}_\Omega$ can be regarded as an orthogonal projection operator in expectation on subspace S.

Proposition 5.3. *Suppose Ω is a set of entries of size m which is sampled independent and uniformly with replacement. Then for any block diagonal matrix $\mathbf{E}=\operatorname{diag}(\mathbf{E}_A,\mathbf{E}_B,\mathbf{E}_C)$ satisfying that $\mathbf{E}\in S$, denoting $\mathcal{O}(\mathbf{E})\triangleq\frac{n_1 n_2 n_3}{m}\mathbb{E}[\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E})]$, we have*

$$\mathcal{P}_S(\mathcal{O}(\mathbf{E}))=\mathbf{E}.$$

Proof. We can calculate $\mathcal{O}(\mathbf{E})$ as follows,

$$\begin{aligned}
\mathcal{O}(\mathbf{E})=&\sum_{abc}\left\langle\mathbf{E},\sigma_{abc}\right\rangle\sigma_{abc}\\
=&\sum_{abc}\left[\left\langle\mathbf{E},\left(\frac{1}{\sqrt{n_3}}\delta_{ab}^{(A)}+\frac{1}{\sqrt{n_1}}\delta_{bc}^{(B)}+\frac{1}{\sqrt{n_2}}\delta_{ca}^{(C)}\right)\right\rangle\right]\cdot\\
&\left(\frac{1}{\sqrt{n_3}}\delta_{ab}^{(A)}+\frac{1}{\sqrt{n_1}}\delta_{bc}^{(B)}+\frac{1}{\sqrt{n_2}}\delta_{ca}^{(C)}\right)
\end{aligned}$$

$$\begin{aligned}
&= \Bigg(\mathbf{E}_A + \frac{1}{\sqrt{n_1 n_3}}\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T\mathbf{E}_B^T + \frac{1}{\sqrt{n_2 n_3}}\mathbf{E}_C^T\mathbf{1}_{n_3}\mathbf{1}_{n_2}^T, \\
&\quad \mathbf{E}_B + \frac{1}{\sqrt{n_1 n_2}}\mathbf{1}_{n_2}\mathbf{1}_{n_1}^T\mathbf{E}_C^T + \frac{1}{\sqrt{n_1 n_3}}\mathbf{E}_A^T\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T, \\
&\quad \mathbf{E}_C + \frac{1}{\sqrt{n_2 n_3}}\mathbf{1}_{n_3}\mathbf{1}_{n_2}^T\mathbf{E}_A^T + \frac{1}{\sqrt{n_1 n_2}}\mathbf{E}_B^T\mathbf{1}_{n_2}\mathbf{1}_{n_1}^T\Bigg) \\
&= \Big(\mathbf{E}_A + \frac{1}{\sqrt{n_1 n_3}}\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T\mathbf{E}_B^T, \mathbf{E}_B + \frac{1}{\sqrt{n_1 n_2}}\mathbf{1}_{n_2}\mathbf{1}_{n_1}^T\mathbf{E}_C^T + \\
&\quad \frac{1}{\sqrt{n_1 n_3}}\mathbf{E}_A^T\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T, \mathbf{E}_C + \frac{1}{\sqrt{n_2 n_3}}\mathbf{1}_{n_3}\mathbf{1}_{n_2}^T\mathbf{E}_A^T\Big),
\end{aligned}$$

where the third equality follows since $\mathbf{1}_{n_2}^T\mathbf{E}_B = \mathbf{0}_{n_3}^T$ and $\mathbf{1}_{n_3}^T\mathbf{E}_C = \mathbf{0}_{n_1}^T$.

Now, by the definition of S_A and $S_A^\perp$, since $\mathbf{1}_{n_3}^T\mathbf{E}_B^T\mathbf{1}_{n_2} = 0$, we have $\mathbf{1}_{n_2}\mathbf{1}_{n_3}^T\mathbf{E}_B^T \in S_A^\perp$ and therefore $\mathcal{P}_{S_A}(\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T\mathbf{E}_B^T) = \mathbf{0}_{n_1\times n_2}$. We also have $\mathcal{P}_{S_B}(\mathbf{1}_{n_2}\mathbf{1}_{n_1}^T\mathbf{E}_C^T) = \mathbf{0}_{n_2\times n_3}$ and $\mathcal{P}_{S_C}(\mathbf{1}_{n_3}\mathbf{1}_{n_2}^T\mathbf{E}_A^T) = \mathbf{0}_{n_3\times n_1}$. In addition, we have $\mathbf{E}_A^T\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T \in S_B^\perp$ and hence $\mathcal{P}_{S_B}(\mathbf{E}_A^T\mathbf{1}_{n_1}\mathbf{1}_{n_3}^T) = \mathbf{0}_{n_3\times n_1}$. Combining these facts, we have

$$\mathcal{P}_S(\mathcal{O}(\mathbf{E})) = \mathbf{E}.$$

□

Next, we show that, with high probability, $\mathcal{R}_\Omega^*\mathcal{R}_\Omega$ is very close to an isometry on subspace T if the number of observations $|\Omega|$ is sufficient by appealing to the non-commutative Bernstein inequality.

Lemma 5.2. *Suppose Ω is a set of entries of size m which is sampled independently and uniformly from $[n_1]\times[n_2]\times[n_3]$ with replacement. Then for all $\beta > 1$,*

$$\frac{n_1 n_2 n_3}{m}\left\|\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T - \frac{m}{n_1 n_2 n_3}\mathcal{P}_T\right\| \le \sqrt{\frac{16 p\mu_0 r n_3 \beta\log(n_3)}{3m}}$$

with probability at least $1 - 2n_3^{2-2\beta}$ if $m > \frac{448}{3}\mu_0 r n_3\beta\log(n_3)$.

Proof. By Proposition 5.3, for any $\mathbf{E} \in T$, we have

$$\begin{aligned}\mathbb{E}\left[\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T\right] &= \mathcal{P}_T\mathbb{E}\left[\mathcal{R}_\Omega^*\mathcal{R}_\Omega\right]\mathcal{P}_T \\ &= \mathcal{P}_T\left(\frac{m}{n_1n_2n_3}\mathcal{O}\right)\mathcal{P}_T \\ &= \frac{m}{n_1n_2n_3}\mathcal{P}_T.\end{aligned}$$

Now we use noncommunicative Bernstein inequality to bound the deviation of the operator $\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T$ from its expected value $\frac{m}{n_1n_2n_3}\mathcal{P}_T$ in spectral norm.

Consider any block diagonal matrix $\mathbf{E} = \operatorname{diag}(\mathbf{E}_A, \mathbf{E}_B, \mathbf{E}_C)$, we can decompose $\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T(\mathbf{E})$ as follows,

$$\begin{aligned}\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T(\mathbf{E}) &= \sum_{k=1}^{m}\left\langle \mathcal{P}_T(\mathbf{E}), \sigma_{a_kb_kc_k}\right\rangle \mathcal{P}_T\left(\sigma_{a_kb_kc_k}\right) \\ &= \sum_{k=1}^{m}\left\langle \mathbf{E}, \mathcal{P}_T(\sigma_{a_kb_kc_k})\right\rangle \mathcal{P}_T(\sigma_{a_kb_kc_k})\end{aligned}$$

Define the operator τ_{abc} which maps $\mathbf{E}$ to $\langle \mathbf{E}, \mathcal{P}_T(\sigma_{abc})\rangle\, \mathcal{P}_T(\sigma_{abc})$. Clearly, we have $\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T = \sum_{k=1}^m \tau_{a_kb_kc_k}$ and $\mathbb{E}[\tau_{a_kb_kc_k}] = \frac{1}{n_1n_2n_3}\mathcal{P}_T$. We can bound the operator norm $\|\tau_{abc}\|$ using Proposition 5.3 as follows

$$\begin{aligned}\|\tau_{abc}\| &= \sup_{\|\mathbf{E}\|_F=1}\|\tau_{abc}(\mathbf{E})\|_F \\ &= \|\mathcal{P}_T(\sigma_{abc})\|_F^2 \\ &\leq \frac{28\mu_0 r}{n_1n_2}.\end{aligned}$$

Now we can compute the bound,

$$\left\|\tau_{a_kb_kc_k} - \frac{1}{n_1n_2n_3}\mathcal{P}_T\right\| \leq \max\left\{\frac{28\mu_0 r}{n_1n_2}, \frac{1}{n_1n_2n_3}\right\} \leq \frac{28\mu_0 r}{n_1n_2},$$

where we have utilized the fact that $\|\mathbf{A} - \mathbf{B}\| \leq \max\{\|\mathbf{A}\|, \|\mathbf{B}\|\}$ for positive semidefinite matrices $\mathbf{A}$ and $\mathbf{B}$. We also have

$$\left\|\mathbb{E}[\tau_{a_kb_kc_k}^2]\right\| = \left\|\mathbb{E}\left[\|\mathcal{P}_T(\sigma_{a_kb_kc_k}))\|_F^2\,\tau_{a_kb_kc_k}\right]\right\|$$

$$\begin{aligned}&\le \frac{28\mu_0 r}{n_1 n_2} \left\|\mathbb{E}\left[\tau_{a_k b_k c_k}\right]\right\| \\ &= \frac{28\mu_0 r}{n_1^2 n_2^2 n_3}.\end{aligned}$$

Therefore,

$$\begin{aligned}\left\|\mathbb{E}\left[\left(\tau_{a_k b_k c_k} - \frac{1}{n_1 n_2 n_3}\mathcal{P}_T\right)^2\right]\right\| &= \left\|\mathbb{E}[\tau^2_{a_k b_k c_k}] - \frac{1}{n_1^2 n_2^2 n_3^2}\mathcal{P}_T\right\| \\ &\le \max\left\{\left\|\mathbb{E}[\tau^2_{a_k b_k c_k}]\right\|, \frac{1}{n_1^2 n_2^2 n_3^2}\right\} \\ &\le \max\left\{\frac{28\mu_0 r}{n_1^2 n_2^2 n_3}, \frac{1}{n_1^2 n_2^2 n_3^2}\right\} \\ &\le \frac{28\mu_0 r}{n_1^2 n_2^2 n_3}.\end{aligned}$$

The lemma follows by applying the noncommuntative Bernstein inequality. □

The next lemma asserts that, for a fixed matrix $\mathbf{E}$, $\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E})$ is close to $\mathcal{O}(\mathbf{E})$ in spectral norm.

Lemma 5.3. *Suppose Ω is a set of entries of size m which is sampled independent and uniformly with replacement. Then, for any $\beta > 1$ and any $\mathbf{E} \in S$,*

$$\left\|\frac{n_1 n_2 n_3}{m}\mathcal{P}_S\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E}\right\| \le \sqrt{\frac{72\beta n_2 n_3^2 \log(n_1+n_2+n_3)}{m}} \left\|\mathbf{E}\right\|_\infty,$$

holds with probability at least $1 - 2(n_1+n_2+n_3)^{1-\beta}$ provided that $m > \frac{98}{9}\beta n_2 \log(n_1+n_2+n_3)$.

Proof. Define the operator γ_{abc} which maps $\mathbf{E}$ to $n_1 n_2 n_3 \left\langle \mathbf{E}, \sigma_{abc}\right\rangle \sigma_{abc}$. We can decompose $\frac{n_1 n_2 n_3}{m}\mathcal{P}_S\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E}$ as

$$\frac{n_1 n_2 n_3}{m}\mathcal{P}_S\mathcal{R}_\Omega^*\mathcal{R}_\Omega - \mathbf{E} = \frac{1}{m}\left[\sum_k \left(\mathcal{P}_S\gamma_{a_k b_k c_k}(\mathbf{E}) - \mathbf{E}\right)\right].$$

We can bound $\|\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E})\|$ as,

$$\begin{aligned}\|\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E})\| &\leq \|\gamma_{a_kb_kc_k}(\mathbf{E})\| \\ &= n_1n_2n_3 \left\|\left\langle \mathbf{E}, \sigma_{a_kb_kc_k}\right\rangle \sigma_{a_kb_kc_k}\right\| \\ &\leq 3n_2n_3 \left\|\mathbf{E}\right\|_\infty\end{aligned}$$

Therefore, we have

$$\begin{aligned}\|\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E}) - \mathbf{E}\| &\leq \|\mathcal{P}_S\gamma_{a_ksb_kc_k}(\mathbf{E})\| + \|\mathbf{E}\| \\ &\leq 3n_2n_3 \left\|\mathbf{E}\right\|_\infty + n_3 \left\|\mathcal{O}(\mathbf{E})\right\|_\infty \\ &\leq \frac{7}{2}n_2n_3 \left\|\mathbf{E}\right\|_\infty,\end{aligned}$$

where we used the fact that $\|\mathbf{E}\| \leq n_3 \left\|\mathbf{E}\right\|_\infty$ and $n \left\|\mathbf{E}\right\|_\infty \leq \frac{1}{2}n_2n_3 \left\|\mathbf{E}\right\|_\infty$ for $n_2 \geq 2$. We also have

$$\begin{aligned}&\left\|\mathbb{E}\left[\left(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E})\right)^*\left(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E})\right)\right]\right\| \\ \leq &\left\|\mathbb{E}\left[\left(\gamma_{a_kb_kc_k}(\mathbf{E})\right)^*\left(\gamma_{a_kb_kc_k}(\mathbf{E})\right)\right]\right\| \\ = &\left\|\mathbb{E}\left[n_1^2n_2^2n_3^2 \left\langle \mathbf{E}, \sigma_{a_kb_kc_k}\right\rangle^2 \sigma^*_{a_kb_kc_k}\sigma_{a_kb_kc_k}\right]\right\| \\ = &\left\|n_1n_2n_3 \sum_{abc} \left\langle \mathbf{E}, \sigma_{abc}\right\rangle^2 \sigma^*_{abc}\sigma_{abc}\right\| \\ \leq &\left\|9n_2n_3 \sum_{abc} \left\|\mathbf{E}\right\|_\infty^2 \sigma^*_{abc}\sigma_{abc}\right\| \\ = &9n_2n_3 \left\|\mathbf{E}\right\|_\infty^2 \left\|\sum_{ab} \delta^{(A)}_{bb} + \sum_{bc} \delta^{(B)}_{bc} + \sum_{ca} \delta^{(C)}_{ca}\right\| \\ \leq &9n_2n_3^2 \left\|\mathbf{E}\right\|_\infty^2.\end{aligned}$$

We now obtain,

$$\begin{aligned}&\left\|\mathbb{E}\left[\left(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E}) - \mathbf{E}\right)^*\left(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E}) - \mathbf{E}\right)\right]\right\| \\ \leq\; &\max\left\{\left\|\mathbf{E}[(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E}))^*(\mathcal{P}_S\gamma_{a_kb_kc_k}(\mathbf{E}))]\right\|, \left\|\mathbf{E}^*\mathbf{E}\right\|\right\} \\ \leq\; &9n_2n_3^2 \left\|\mathbf{E}\right\|_\infty^2.\end{aligned}$$

Then the lemma follows by the noncommunative Bernstein Inequality. □

The next concentration result is the final piece for constructing the dual certificate.

Lemma 5.4. *Suppose Ω is a set of entries of size m sampled independently with replacement. Then for any $\mathbf{E} \in \mathcal{T}$ and any $\beta > 2$, we have*

$$\left\| \frac{n_1 n_2 n_3}{m} \mathcal{P}_T \mathcal{R}_\Omega^* \mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E} \right\|_\infty \leq \sqrt{\frac{864 \beta \mu_0 r n_3 \log n_3}{m}} \left\| \mathbf{E} \right\|_\infty$$

with probability at least $1 - 6 n_3^{2-\beta}$ if $m > 50 \beta \mu_0 r n_3 \log(n_3)$.

Proof. Without loss of generality, for each a, b, we define the random variable

$$\chi_{ab^{(A)}} = \left\langle \delta_{ab}^{(A)}, n_1 n_2 n_3 \left\langle \mathbf{E}, \sigma_{a'b'c'} \right\rangle \mathcal{P}_T(\sigma_{a'b'c'}) - \mathbf{E} \right\rangle,$$

where a', b', c' is sampled uniformly random from $[n_1] \times [n_2] \times [n_3]$. We also define $\chi_{bc^{(B)}}$ and $\chi_{ca^{(C)}}$ similarly. We now bound each of $\chi_{ab^{(A)}}$, $\chi_{bc^{(B)}}$ and $\chi_{ca^{(C)}}$ using standard Berstein inequality. By definition, we have $\mathbb{E}[\chi_{ab^{(A)}}] = 0$ and

$$\begin{aligned}
|\chi_{ab^{(A)}}| &\leq \left| \left\langle \delta_{ab}^{(A)}, n_1 n_2 n_3 \left\langle \mathbf{E}, \sigma_{a'b'c'} \right\rangle \mathcal{P}_T(\sigma_{a'b'c'}) \right\rangle \right| + \left| \left\langle \delta_{ab}^{(A)}, \mathbf{E} \right\rangle \right| \\
&= n_1 n_2 n_3 \left| \left\langle \mathbf{E}, \sigma_{a'b'c'} \right\rangle \right| \left| \left\langle \delta_{ab}^{(A)}, \mathcal{P}_T(\sigma_{a'b'c'}) \right\rangle \right| + \left| \left\langle \delta_{ab}^{(A)}, \mathbf{E} \right\rangle \right| \\
&\leq 3 \sqrt{n_1} n_2 n_3 \left\| \mathbf{E} \right\|_\infty \left\| \mathcal{P}_T(\delta_{ab}^{(A)}) \right\|_F \left\| \mathcal{P}_T(\sigma_{a'b'c'}) \right\|_F + \left\| \mathbf{E} \right\|_\infty \\
&\leq 90 n_3 \mu_0 r \left\| \mathbf{E} \right\|_\infty
\end{aligned}$$

Now we can also bound $\mathbb{E}[\chi^2_{ab^{(A)}}]$ as follows,

$$\begin{aligned}
\mathbb{E}[\chi^2_{ab^{(A)}}] &= \frac{1}{n_1 n_2 n_3} \sum_{a'b'c'} \left\langle \delta_{ab}^{(A)}, n_1 n_2 n_3 \left\langle \mathbf{E}, \sigma_{a'b'c'} \mathcal{P}_T(\sigma_{a'b'c'}) \right\rangle - \mathbf{E} \right\rangle^2 \\
&= n_1 n_2 n_3 \sum_{a'b'c'} \left\langle \mathbf{E}, \sigma_{a'b'c'} \right\rangle^2 \left\langle \delta_{ab}^{(A)}, \mathcal{P}_T(\sigma_{a'b'c'}) \right\rangle^2 - \left\langle \mathbf{E}, \delta_{ab}^{(A)} \right\rangle^2 \\
&\leq n_1 n_2 \sum_{a'b'c'} \left\langle \mathbf{E}, \sigma_{a'b'c'} \right\rangle^2 \left\langle \delta_{ab}^{(A)}, \mathcal{P}_T(\delta_{a'b'}^{(A)}) \right\rangle^2
\end{aligned}$$

$$\begin{aligned}
&\leq 9n_1n_2n_3 \|\mathbf{E}\|_\infty^2 \sum_{a'b'} \left\langle \delta_{ab}^{(A)}, \mathcal{P}_T(\sigma_{a'b'}) \right\rangle^2 \\
&\leq 9n_1n_2n_3 \|\mathbf{E}\|_\infty^2 \left\| \mathcal{P}_T(\delta_{ab}^{(A)}) \right\|_F^2 \\
&\leq 162\mu_0 r n_3 \|\mathbf{E}\|_\infty^2 .
\end{aligned}$$

Clearly the entry $\left\langle \delta_{ab}^{(A)}, \frac{n_1n_2n_3}{m}\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E} \right\rangle$ is the mean value of m i.i.d copies of $\chi_{ab^{(A)}}$. Apply the Bernstein's Inequality, we have

$$\Pr\left[\left| \left\langle \delta_{ab}^{(A)}, \frac{n_1n_2n_3}{m}\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E} \right\rangle \right| > \sqrt{\frac{864\beta\mu_0 r n_3 \log(n_3)}{m}} \|\mathbf{E}\|_\infty \right] \\ \leq 2n_3^{-\beta}.$$

By union bound, we have

$$\Pr\left[\left\| \frac{n_1n_2n_3}{m}\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{E}) - \mathbf{E} \right\|_\infty > \sqrt{\frac{864\beta\mu_0 r n_3 \log(n_3)}{m}} \|\mathbf{E}\|_\infty \\ \leq 6n_3^{2-\beta}.$$

□

Finally, we adapt the "golfling scheme" proposed by [68] to construct the dual certificate $\mathbf{F}$.

Lemma 5.5. *Suppose Ω is a set of entries of size sample independently with replacement for $m > 3600 \max\{\mu_1^2, \mu_0\} r n_3 \beta \log^2(6n_3)$. There exists $\mathbf{F} \in \operatorname{range}(\mathcal{R}_\Omega^*)$ satisfies*

$$\|\mathcal{P}_T(\mathbf{F}) - \mathbf{W}\|_F \leq \sqrt{\frac{r}{2n_3}}, \quad \|\mathcal{P}_{T^\perp}(\mathbf{F})\| < \frac{1}{2},$$

with probability at least $1 - 3\log(6n_3)(3n_3)^{2-\beta}$ for all $\beta > 2$.

Proof. Partition m entries of Ω into p partitions of size q, where

$$q \geq 3600 \max\{\mu_0, \mu_1^2\} r n_3 \beta \log(6n_3), \quad p \geq \log(6n_3).$$

Denote Ω_j be the j th partition. By Lemma 5.2 and union bound, we have

$$\Pr\Big[\frac{n_1n_2n_3}{q}\Big\|\mathcal{P}_T\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}\mathcal{P}_T-\frac{q}{n_1n_2n_3}\mathcal{P}_T\leq\frac{1}{2}\quad\text{for all } j\in[p]\Big]$$
$$\geq 1-\log(6n_3)2n_3^{2-2\beta}.$$

Now suppose the above event happens. Define $\mathbf{F}_0=0$, $\mathbf{G}_0=\mathbf{W}$ and

$$\mathbf{F}_j=\mathbf{F}_{j-1}+\frac{n_1n_2n_3}{q}\mathcal{R}^*_{\Omega_{j-1}}\mathcal{R}_{\Omega_{j-1}}(\mathbf{G}_{j-1}),\mathbf{G}_j=\mathbf{W}-\mathcal{P}_T(\mathbf{F}_j)$$

for $j\in[p]$. We can now bound $\|\mathbf{G}_j\|_F$ as follows,

$$\begin{aligned}\|\mathbf{G}_j\|_F&=\|\mathbf{W}-\mathcal{P}_T(\mathbf{F}_j)\|_F\\&=\left\|\mathbf{W}-\mathcal{P}_T(\mathbf{F}_{j-1})-\frac{n_1n_2n_3}{q}\mathcal{P}_T\mathcal{R}^*_{\Omega_{j-1}}\mathcal{R}_{\Omega_{j-1}}(\mathbf{G}_{j-1})\right\|_F\\&=\left\|\mathbf{G}_{j-1}-\frac{n_1n_2n_3}{q}\mathcal{P}_T\mathcal{R}^*_{\Omega_{j-1}}\mathcal{R}_{\Omega_{j-1}}(\mathbf{G}_{j-1})\right\|_F\\&\leq\frac{1}{2}\|\mathbf{G}_{j-1}\|_F.\end{aligned}$$

It follows that $\|\mathbf{G}_p\|_F\leq 2^{-p}\|\mathbf{G}_0\|_F=2^{-p}\sqrt{3r}\leq\frac{r}{2n_3}$, since $p\geq\log(2n_3)\geq\log_2\sqrt{2n_3}$. Now choose $\mathbf{F}=\mathbf{F}_p$, it is easy to check that

$$\|\mathcal{P}_T(\mathbf{F})-\mathbf{W}\|_F\leq\sqrt{\frac{r}{2n_3}}$$

with at least probability $1-\log(6n_3)2n_3^{2-\beta}$.

We now argue that $\mathbf{F}_p$ also satisfies the second inequality in this lemma with high probability. Apply Lemma 5.3 and Lemma 5.4, we have

$$\Pr\Big[\left\|\frac{n_1n_2n_3}{q}\mathcal{P}_S\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}(\mathbf{G}_{j-1})-\mathbf{G}_{j-1}\right\|\leq$$
$$\sqrt{\frac{72n_1n_2^2\beta\log(n_1+n_2+n_3)}{q}}\|\mathbf{G}_{j-1}\|_\infty\Big]\geq 1-2(n_1+n_2+n_3)^{1-\beta},$$

$$\Pr\left[\left\|\mathbf{G}_{j-1} - \frac{n_1n_2n_3}{q}\mathcal{P}_T\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}(\mathbf{G}_{j-1})\right\| \le \frac{1}{2}\left\|\mathbf{G}_{j-1}\right\|_\infty\right] \ge 1-6n_3^{2-\beta}.$$

By union bound, the above random events holds for all $j = 1,\ldots,p$ with probability at least $1-2\log(6n_3)(3n_3)^{1-\beta}$. Suppose these random event happens, we can bound $\mathcal{P}_{T^\perp}(\mathbf{F}_p)$ as follows.

$$\begin{aligned}
\|\mathcal{P}_{T^\perp}(\mathbf{F}_p)\| &\le \sum_{j=1}^{p}\left\|\frac{n_1n_2n_3}{q}\mathcal{P}_{T^\perp}\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}(\mathbf{G}_{j-1})\right\| \\
&= \sum_{j=1}^{p}\left\|\mathcal{P}_{T^\perp}\left(\frac{n_1n_2n_3}{q}\mathcal{P}_S\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}(\mathbf{G}_{j-1}) - \mathbf{G}_{j-1}\right)\right\| \\
&\le \sum_{j=1}^{p}\left\|\frac{n_1n_2n_3}{q}\mathcal{P}_S\mathcal{R}^*_{\Omega_j}\mathcal{R}_{\Omega_j}(\mathbf{G}_{j-1}) - \mathbf{G}_{j-1}\right\| \\
&\le \sum_{j=1}^{p}\sqrt{\frac{72n_2n_3^2\beta\log(n_1+n_2+n_3)}{q}}\left\|\mathbf{G}_{j-1}\right\|_\infty \\
&= 2\sum_{j=1}^{p}2^{-j}\sqrt{\frac{72n_2n_3^2\beta\log(n_1+n_2+n_3)}{q}}\left\|\mathbf{W}\right\|_\infty \\
&< \sqrt{\frac{288\mu_1^2rn_3\beta\log(3n_3)}{q}} \\
&< \frac{1}{2},
\end{aligned}$$

where the second equality holds because $\mathcal{P}_{T^\perp}(\mathbf{G}_{j-1}) = \mathbf{0}$ for all j and the last inequality follows since $q > 3600\mu_1^2rn_3\beta\log(6n_3)$. Finally, by union bound, the probability that all above random events happen is at least $1-3\log(6n_3)(3n_3)^{2-\beta}$. □

Remark 5.1. *By golfing scheme construction, the dual certificate* $\mathbf{F}$ *can be decomposed by*

$$\mathbf{F} = \sum_{i=1}^{p}\frac{n_1n_2n_3}{q}\mathcal{R}^*_{\Omega_i}\mathcal{R}_{\Omega_i}(\mathbf{G}_i),$$

for some $\mathbf{G}_1, \ldots, \mathbf{G}_p$.

We will use this property later for proving the stable recovery result in the presence of noise.

The next lemma is an elementary Chernoff bound which shows that maximum duplication of any entry in Ω when sampling with replacement is bounded by $\frac{8}{3}\beta \log(n_1)$. This gives us the upper bound of the spectral norm of $\mathcal{R}_\Omega$.

Lemma 5.6. *Suppose* Ω *is a set of entries of size sample independently with replacement for* $m > 3600 \max\{\mu_1^2, \mu_0\} r n_3 \beta \log^2(6n_3)$. *We have*

$$\|\mathcal{R}_\Omega\| \le \sqrt{\frac{8\beta \log(n_3)}{n_1}}$$

for $n_3 \ge 1$ *and* $\beta \ge 1$ *with probability at least* $1 - 3n_3^{2-2\beta}$.

Proof. Given a set of entries $\Omega = \{(a_k, b_k, c_k)\}_{k\in[m]}$ sampled uniformly with replacement, denote the number of reptitions as $\eta_{ab}^{(A)} = |\{k | a_k = a, b_k = b\}|$, $\eta_{bc}^{(B)} = |\{k | b_k = b, c_k = c\}|$ and $\eta_{ca}^{(C)} = |\{k | c_k = c, a_k = a\}|$.

$$\begin{aligned}
&\|\mathcal{R}_\Omega\| \\
=& \sup_{\|\mathbf{E}\|_F = 1} \|\mathcal{R}_\Omega(\mathbf{E})\|_F \\
\le& \frac{1}{\sqrt{n_1}} \sup_{\|\mathbf{E}\|_F = 1} \sqrt{\sum_{k=1}^m \left\langle \mathbf{E}, \delta_{a_k b_k}^{(A)} + \delta_{b_k c_k}^{(B)} + \delta_{c_k a_k}^{(C)} \right\rangle^2} \\
\le& \frac{1}{\sqrt{n_1}} \sup_{\|\mathbf{E}\|_F = 1} \sqrt{3 \left[\sum_{k=1}^m \left\langle \mathbf{E}, \delta_{a_k b_k}^{(A)} \right\rangle^2 + \sum_{k=1}^m \left\langle \mathbf{E}, \delta_{b_k c_k}^{(B)} \right\rangle^2 + \sum_{k=1}^m \left\langle \mathbf{E}, \delta_{c_k a_k}^{(C)} \right\rangle^2 \right]} \\
=& \frac{1}{\sqrt{n_1}} \sup_{\|\mathbf{E}\|_F = 1} \sqrt{3 \left[\sum_{ab} \left\langle \mathbf{E}, \delta_{ab}^{(A)} \right\rangle^2 \eta_{ab}^{(A)} + \sum_{bc} \left\langle \mathbf{E}, \delta_{bc}^{(B)} \right\rangle^2 \eta_{bc}^{(B)} + \sum_{ca} \left\langle \mathbf{E}, \delta_{ca}^{(C)} \right\rangle^2 \eta_{ca}^{(C)} \right]} \\
=& \frac{1}{\sqrt{n_1}} \sqrt{3 \max\{\max_{ab} \eta_{ab}^{(A)}, \max_{bc} \eta_{bc}^{(B)}, \max_{ca} \eta_{ca}^{(C)}\}}.
\end{aligned}$$

Therefore, it suffices to bound the maximum number of repetitions of any entry in Ω. To this end, we can apply Chernoff bound for the Bernoulli distribution. The probability of an entry a, b be sampled for more than t times can be bounded by Chernoff bound.

$$\Pr\left[\eta_{ab}^{(A)} \geq \frac{8}{3}\beta \log(n_1)\right] \leq \left(\frac{8}{3}\beta \log(n_1)\right)^{-\frac{8}{3}\beta\log(n_1)} \exp\left(\frac{8}{3}\beta \log(n_1)\right)$$
$$\leq n_1^{-2\beta},$$

when $n_1 \geq 9$. We can also bound $\eta_{bc}^{(B)}$ and $\eta_{ca}^{(C)}$ similarly. By union bound, we have

$$\max\{\max_{ab} \eta_{ab}^{(A)}, \max_{bc} \eta_{bc}^{(B)}, \max_{ca} \eta_{ca}^{(C)}\} \leq \frac{8}{3}\beta \log(n)$$

hold with probability at least $1 - 3n_3^{2-2\beta}$ by union bound. □

Finally, to apply Theorem 5.4, we require the following bound relating $\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_*$ and $\|\mathcal{P}_T(\mathbf{E})\|_F$ for any fixed matrix $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$.

Lemma 5.7. *Suppose Ω is a set of entries of size sample independently with replacement for $m > 3600 \max\{\mu_1^2, \mu_0\} r n_3 \beta \log^2(2n_3)$. Then, for any $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$, we have*

$$\frac{1}{2}\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* < \sqrt{\frac{r}{2n_3}}\|\mathcal{P}_T(\mathbf{E})\|_F,$$

with probability at least $1 - 3n_3^{2-\beta}$.

Proof. Since $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$, we have

$$0 = \|\mathcal{R}_\Omega(\mathbf{E})\|_F \geq \|\mathcal{R}_\Omega\mathcal{P}_T(\mathbf{E})\|_F - \|\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(\mathbf{E})\|_F.$$

Apply Lemma 5.2,

$$\frac{n_1 n_2 n_3}{m}\left\|\mathcal{P}_T\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_T - \frac{m}{n_1 n_2 n_3}\mathcal{P}_T\right\| \leq \frac{1}{2} \tag{5.16}$$

holds with probability at least $1 - 3n_3^{2-\beta}$. Suppose Eq. (5.16) holds, we can bound $\|\mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E})\|_F$ as follows

$$\begin{aligned}\|\mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E})\|_F^2 &= \langle \mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E}), \mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E})\rangle \\ &= \langle \mathbf{E}, \mathcal{P}_T \mathcal{R}_\Omega^* \mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E})\rangle \\ &\geq \frac{m}{2n_1n_2n_3} \|\mathcal{P}_T(\mathbf{E})\|_F^2 .\end{aligned}$$

On the other hand, we need to bound $\|\mathcal{R}_\Omega \mathcal{P}_{T^\perp}(\mathbf{E})\|_F$. Suppose $\|\mathcal{R}_\Omega\| \leq \sqrt{\frac{8\beta \log(n_3)}{n_1}} \leq \sqrt{\frac{8\beta}{n_1}} \log(n_3)$ which holds with probability at least $1 - n_3^{2-\beta}$ by Lemma 5.6, we have

$$\begin{aligned}\|\mathcal{R}_\Omega \mathcal{P}_{T^\perp}(\mathbf{E})\|_F &\leq \|\mathcal{R}_\Omega\| \, \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_F \\ &\leq \sqrt{\frac{8\beta}{n_1}} \log(n_3) \, \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_F .\end{aligned}$$

Therefore,

$$\begin{aligned}\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* &\geq \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_F \\ &\geq \sqrt{\frac{n_1}{8\beta \log^2(n_3)}} \, \|\mathcal{R}_\Omega \mathcal{P}_{T^\perp}(\mathbf{E})\|_F \\ &\geq \sqrt{\frac{n_1}{8\beta \log^2(n_3)}} \, \|\mathcal{R}_\Omega \mathcal{P}_T(\mathbf{E})\|_F \\ &\geq \sqrt{\frac{mn_1}{8n_1n_2n_3\beta \log^2(n_3)}} \, \|\mathcal{P}_T(\mathbf{E})\|_F \\ &\geq \sqrt{\frac{3600n_3n_1r\mu_0\beta \log^2(6n_3)}{8n_1n_2n_3\beta \log^2(6n)}} \, \|\mathcal{P}_T(\mathbf{E})\|_F \\ &> \sqrt{\frac{2r}{n_3}} \, \|\mathcal{P}_T(\mathbf{E})\|_F .\end{aligned}$$

□

We are now ready to prove the exact recovery result Theorem 5.1.

Proof. (Theorem 1) By Lemma 5.5, there exists $\mathbf{F} \in \text{range}(\mathcal{R}_\Omega^*)$ such that

$$\|\mathcal{P}_T(\mathbf{F}) - \mathbf{W}\|_F \leq \sqrt{\frac{r}{2n_3}}, \quad \|\mathcal{P}_{T^\perp}(\mathbf{F})\| < \frac{1}{2},$$

with probability at least $1 - 3\log(6n_3)(3n_3)^{2-\beta}$ for all $\beta > 2$.

On the other hand, Lemma 5.7 shows that for any $\mathbf{E} \in \ker(\mathcal{R}_\Omega)$,

$$\frac{1}{2}\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* > \sqrt{\frac{r}{2n_3}}\|\mathcal{P}_T(\mathbf{E})\|_F\,,$$

holds with probability at least $1 - 3n_3^{2-\beta}$.

By union bound, the above random events happen simultaneously with probability at least $1 - 3\log(6n_3)(3n_3)^{2-\beta} - 3n_3^{2-\beta}$. Finally, in the case that both random events holds, by Theorem. 5.4, the solution to Eq. (5.3) is unique and exactly recovers $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and therefore the pairwise interaction tensor $\mathcal{T} = \text{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$. □

5.7 Proof of Theorem 5.2

In this section, we generalize the proof of Theorem 5.2 for the noisy cases.

Proof. (Theorem 2) First, define operator $\mathcal{Q} = \gamma^2\mathcal{R}_\Omega^*\mathcal{R}_\Omega$, where $\gamma = \|\mathcal{R}_\Omega\|^{-1}$, as the normalized version of $\mathcal{R}_\Omega^*\mathcal{R}_\Omega$. Clearly, we have $\|\mathcal{Q}\| = 1$. By Lemma 5.6, we can bound γ by $\gamma \geq \sqrt{\frac{n_1}{8\beta\log(n_3)}}$.

We can decompose the optimal solution $\hat{\mathbf{M}}$ of the convex progam Eq. (5.5) into the sum of the true matrix $\mathbf{M}$ and the error matrix $\mathbf{E}$, namely, $\hat{\mathbf{M}} = \mathbf{M} + \mathbf{E}$. To prove the theorem, we need to bound the error term $\mathbf{E}$ in its nuclear norm $\|\mathbf{E}\|_*$. To do this, we start with bounding $\|\mathcal{Q}(\mathbf{E})\|_F$. Denote the noisy observations as an m-dimensional vector $\mathbf{y}$, where $y_i = T_{a_ib_ic_i}$.

We have

$$\begin{aligned}
&\|\mathcal{Q}(\mathbf{E})\|_F && (5.17)\\
\leq&\left\|\mathcal{Q}(\hat{\mathbf{M}})-\gamma^2\mathcal{R}_\Omega^*(\mathbf{y})\right\|_F+\left\|\gamma^2\mathcal{R}_\Omega^*(\mathbf{y})-\mathcal{Q}(\mathbf{M})\right\|_F\\
=&\gamma^2\left\|\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\hat{\mathbf{M}})-\mathcal{R}_\Omega^*(\mathbf{y})\right\|_F+\gamma^2\left\|\mathcal{R}_\Omega^*(\mathbf{y})-\mathcal{R}_\Omega^*\mathcal{R}_\Omega(\mathbf{M})\right\|_F\\
\leq&\gamma^2\left\|\mathcal{R}_\Omega^*\right\|\left\|\mathcal{R}_\Omega(\hat{\mathbf{M}})-\mathbf{y}\right\|_F+\gamma^2\left\|\mathcal{R}_\Omega^*\right\|\left\|\mathbf{y}-\mathcal{R}_\Omega(\mathbf{M})\right\|_F\\
\leq&\gamma\epsilon_1+\gamma\epsilon_2\triangleq\delta. && (5.18)
\end{aligned}$$

In the last inequality, the first term $\left\|\mathcal{R}_\Omega(\hat{\mathbf{M}})-\mathbf{y}\right\|_F$ is no greater than ϵ_2 due to constraint of optimization problem and the second term $\|\mathbf{y}-\mathcal{R}_\Omega(\mathbf{M})\|_F\leq\epsilon_1$ is the assumption on the observation noise.

On the other hand, we can bound $\|\mathcal{Q}(\mathbf{E})\|_F$ by following

$$\|\mathcal{Q}(\mathbf{E})\|_F\geq\|\mathcal{Q}\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F-\|\mathcal{Q}\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\|_F.$$

For the second term, we have $\|\mathcal{Q}\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\|_F\leq\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\|_F$. Now, we focus on the first term, we have

$$\begin{aligned}
&\|\mathcal{Q}\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F\\
=&\gamma^2\|\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F\\
\geq&\gamma^2\|\mathcal{P}_\mathcal{T}\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F\\
\geq&\gamma^2\frac{m}{n_1n_2n_3}\left\|\frac{n_1n_2n_3}{m}\mathcal{P}_\mathcal{T}\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{E})\right\|_F\\
\geq&\gamma^2\frac{m}{n_1n_2n_3}\left[\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F-\left\|\frac{n_1n_2n_3}{m}\mathcal{P}_\mathcal{T}\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{E})-\mathcal{P}_\mathcal{T}(\mathbf{E})\right\|_F\right]\\
\geq&\gamma^2\frac{m}{n_1n_2n_3}\left[\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F-\left\|\frac{n_1n_2n_3}{m}\mathcal{P}_\mathcal{T}\mathcal{R}_\Omega^*\mathcal{R}_\Omega\mathcal{P}_\mathcal{T}-\mathcal{P}_\mathcal{T}\right\|\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F\right]\\
\geq&\gamma^2\frac{m}{n_1n_2n_3}\frac{1}{2}\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F\geq\frac{m}{16\beta\log(n_3)n_2n_3}\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F
\end{aligned}$$

Therefore, we have

$$\|\mathcal{Q}(\mathbf{E})\|_F\geq\frac{m}{16\beta n_2n_3\log(n_3)}\|\mathcal{P}_\mathcal{T}(\mathbf{E})\|_F-\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\|_F. \quad (5.19)$$

Now, combine Eq. (5.18) and Eq. (5.19), we have

$$\begin{aligned}\left\|\mathcal{P}_{\mathcal{T}}(\mathbf{E})\right\|_F &\leq \frac{16\beta n_2 n_3 \log(n_3)}{m}\left(\delta + \left\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\right\|_F\right)\\ &\leq \frac{16\beta n_2 n_3 \log(n_3)}{m}\left(\delta + \left\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\right\|_*\right).\end{aligned}$$

Next, we proceed to bound $\left\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\right\|_*$. We can use a similar subgradient argument as in the proof of Theorem 5.4. Let $\mathbf{F} \in \text{range}(\mathcal{R}_\Omega^*)$ be the dual certificate as described in Theorem 5.4, we have

$$\begin{aligned}&\left\|\hat{\mathbf{M}}\right\|_*\\ =&\left\|\mathbf{M}+\mathbf{E}\right\|_*\\ \geq&\left\langle \mathbf{W}+\mathbf{W}_\perp, \mathbf{M}+\mathbf{E}\right\rangle = \left\|\mathbf{M}\right\|_* + \left\langle \mathbf{W}+\mathbf{W}_\perp, \mathbf{E}\right\rangle\\ =&\left\|\mathbf{M}\right\|_* + \left\langle \mathbf{W}+\mathbf{W}_\perp - \mathbf{F}, \mathbf{E}\right\rangle + \left\langle \mathbf{F}, \mathbf{E}\right\rangle\\ =&\left\|\mathbf{M}\right\|_* + \left\langle \mathbf{W} - \mathcal{P}_T(\mathbf{F}), \mathcal{P}_{\mathcal{T}}(\mathbf{E})\right\rangle + \left\langle \mathbf{W}_\perp - \mathcal{P}_{\mathcal{T}^\perp}(\mathbf{F}), \mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\right\rangle + \left\langle \mathbf{F}, \mathbf{E}\right\rangle\\ \geq&\left\|\mathbf{M}\right\|_* - \frac{\sqrt{r}}{n_3^2}\left\|\mathcal{P}_{\mathcal{T}}(\mathbf{E})\right\|_F + \frac{1}{2}\left\|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{E})\right\|_* + \left\langle \mathbf{F}, \mathbf{E}\right\rangle.\end{aligned}$$

Recall that $\hat{\mathbf{M}}$ is the optimal solution to the convex program Eq. (5.5), we have $\left\|\hat{\mathbf{M}}\right\|_* \leq \left\|\mathbf{M}\right\|_*$. Hence, we have

$$\frac{1}{2}\left\|\mathcal{P}_{T^\perp}(\mathbf{E})\right\|_* \leq \frac{\sqrt{r}}{n_3^2}\left\|\mathcal{P}_T(\mathbf{E})\right\|_F - \left\langle \mathbf{F}, \mathbf{E}\right\rangle.$$

Now we bound $\langle \mathbf{F}, \mathbf{E}\rangle$. By the golfling scheme construction of $\mathbf{F}$ in Lemma. 5.5, we have

$$\begin{aligned}\left\langle \mathbf{F}, \mathbf{E}\right\rangle &= \sum_{i=1}^{p}\left\langle \mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{G}_i), \mathbf{E}\right\rangle\\ &= \sum_{i=1}^{p}\left\langle \mathbf{G}_i, \mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{E})\right\rangle\\ &\geq -\sum_{i=1}^{p}\left\|\mathbf{G}_i\right\|_F\left\|\mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{E})\right\|_F.\end{aligned}$$

For each i, we can bound $\left\|\mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{E})\right\|_F$ by

$$\begin{aligned}\left\|\mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{E})\right\|_F &= \left\|\mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\hat{\mathbf{M}}) - \mathcal{R}_{\Omega_i}^*(\mathbf{y}_{\Omega_i})\right\|_F + \left\|\mathcal{R}_{\Omega_i}^*(\mathbf{y}_{\Omega_i}) - \mathcal{R}_{\Omega_i}^*\mathcal{R}_{\Omega_i}(\mathbf{M})\right\|_F \\ &\leq \left\|\mathcal{R}_{\Omega_i}^*\right\| \left\|\mathcal{R}_{\Omega_i}(\hat{\mathbf{M}}) - \mathbf{y}_{\Omega_i}\right\|_F + \left\|\mathcal{R}_{\Omega_i}^*\right\| \left\|\mathbf{y}_{\Omega_i} - \mathcal{R}_{\Omega_i}(\mathbf{M})\right\|_F \\ &= \frac{\epsilon_1}{\gamma} + \frac{\epsilon_2}{\gamma},\end{aligned}$$

where $\mathbf{y}_{\Omega_i}$ is the restriction of $\mathbf{y}$ on Ω_i.

Therefore, we have

$$\begin{aligned}\langle \mathbf{F}, \mathbf{E}\rangle &\geq -\frac{\epsilon_1+\epsilon_2}{\gamma}\sum_{i=1}^{p}\|\mathbf{G}_i\|_F \\ &\geq -\frac{2(\epsilon_1+\epsilon_2)}{\gamma}\|\mathbf{G}_0\|_F \\ &= -\frac{2(\epsilon_1+\epsilon_2)}{\gamma}\|\mathbf{W}\|_F \geq -\frac{2(\epsilon_1+\epsilon_2)}{\gamma}\sqrt{r_1+r_2+r_3}.\end{aligned}$$

Consequently, for reasonable values of parameters, we have

$$\begin{aligned}\frac{1}{2}\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* &\leq \frac{\sqrt{r}}{n_3^2}\|\mathcal{P}_T(\mathbf{E})\|_F + \frac{2(\epsilon_1+\epsilon_2)}{\gamma}\sqrt{r_1+r_2+r_3} \\ &\leq \frac{16\beta\log(n_3)\sqrt{r}}{m}(\delta + \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_*) + \frac{2(\epsilon_1+\epsilon_2)}{\gamma}\sqrt{r_1+r_2+r_3} \\ &\leq \frac{1}{16}(\delta + \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_*) + \frac{\epsilon_1+\epsilon_2}{16}.\end{aligned}$$

Hence, we have

$$\|\mathcal{P}_{T^\perp}(\mathbf{E})\|_* \leq \frac{16}{7}\left(\delta + \frac{\epsilon_1+\epsilon_2}{16}\right) \leq 3\delta.$$

Last, combining the above inequalities and setting $\epsilon = \epsilon_1 + \epsilon_2$, we can finally bound the error $\mathbf{E}$ in terms of its nuclear norm as follows

$$\|\mathbf{E}\|_* \leq \sqrt{2r}\,\|\mathcal{P}_T(\mathbf{E})\|_F + \|\mathcal{P}_{T^\perp}(\mathbf{E})\|_*$$

$$\begin{aligned}
&\leq \sqrt{2r}n_2\delta + (\sqrt{2r}n_2 + 1)\left\|\mathcal{P}_{T^\perp}(\mathbf{E})\right\|_* \\
&\leq \sqrt{2r}n_2\gamma\epsilon + 3(\sqrt{2r}n_2 + 1)\gamma\epsilon \\
&\leq 5\sqrt{2r}n_2\gamma\epsilon \\
&\leq 5\sqrt{\frac{2rn_1n_2^2}{8\beta\log(n_1)}}\epsilon.
\end{aligned}$$

□

5.8 Proof of Proposition 5.1

We first review some terminologies. We call a matrix $\mathbf{A}$ a *doubly centered matrix*, if each column and each row of $\mathbf{A}$ sums up to zero, i.e. $\mathbf{1}^T\mathbf{A} = \mathbf{0}^T$ and $\mathbf{A}\mathbf{1} = \mathbf{0}$ hold simultaenously. We also call a vector $\mathbf{v}$ a *centered vector*, if the sum of its entries equals to zero, namely, $\mathbf{1}^T\mathbf{v} = 0$.

Lemma 5.8. *Given an arbitrary pariwise interaction tensor* $\mathcal{T} = \mathrm{Pair}(\mathbf{A}, \mathbf{B}, \mathbf{C})$*, there exists a unique 7-tuple* $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ *such that* $\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0$ *are doubly centered matrices,* $\mathbf{a}, \mathbf{b}, \mathbf{c}$ *are centered vectors and satisfies*

$$T_{ijk} = A^0_{ij}+B^0_{jk}+C^0_{ki}+a_i+b_j+c_k+d, \quad \textit{for all } (i,j,k) \in [n_1]\times[n_2]\times[n_3]. \tag{5.20}$$

Remark 5.2. *We can interpret the quantities* $\mathbf{a}, \mathbf{b}, \mathbf{c}, d$ *in Lemma 5.8 as axis-aligned biases of tensor* $\mathcal{T}$*. For example, every entries of the form* T_{1jk} *are influenced by bias* a_1*; the entries of form* T_{i1k} *for all* (i,k) *are biased by* b_1*; the entries of form* T_{ij1} *for all* (i,j) *are biased by* c_1*. In addition, all entries of* $\mathcal{T}$ *is biased by* d*.*

Proof. (Lemma 5.8) In the following, we shall prove the existence and uniqueness separately.

Existence. Given any $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ of appropriate size, we now construct the 7-tuple $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ specified in the lemma.

We define the mean values of matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ by $\sigma_A = \frac{1}{n_1 n_2}\mathbf{1}^T\mathbf{A}\mathbf{1}$, $\sigma_B = \frac{1}{n_2 n_3}\mathbf{1}^T\mathbf{B}\mathbf{1}$ and $\sigma_C = \frac{1}{n_3 n_1}\mathbf{1}^T\mathbf{B}\mathbf{1}$. We also denote the mean vectors of columns of matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ by $\mathbf{a}_c = \frac{1}{n_1}\mathbf{A}^T\mathbf{1}$, $\mathbf{b}_c = \frac{1}{n_2}\mathbf{B}^T\mathbf{1}$ and $\mathbf{c}_c = \frac{1}{n_3}\mathbf{C}^T\mathbf{1}$. Similarly, we denote the mean vectors of rows of matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ by $\mathbf{a}_r = \frac{1}{n_2}\mathbf{A}\mathbf{1}$, $\mathbf{b}_r = \frac{1}{n_3}\mathbf{B}\mathbf{1}$ and $\mathbf{c}_r = \frac{1}{n_1}\mathbf{C}\mathbf{1}$.

Now, we construct the desired 7-tuple $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ by

$$A^0_{ij} = A_{ij} - a^c_j - a^r_i + \sigma_a, \quad B^0_{jk} = B_{jk} - b^c_j - b^r_k + \sigma_b, \quad C^0_{ki} = C_{ki} - c^c_i - c^r_k + \sigma_c$$
$$a_i = a^r_i + c^c_i - \sigma_a - \sigma_c, \quad b_j = b^r_j + a^c_j - \sigma_b - \sigma_a, \quad c_k = c^r_k + b^c_k - \sigma_c - \sigma_b$$
$$d = \sigma_a + \sigma_b + \sigma_c,$$

where (i, j, k) ranges within $[n_1] \times [n_2] \times [n_3]$. It is easy to verify that $\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0$ are doubly centered matrices and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are centered vectors and that $A_{ij} + B_{jk} + C_{ki} = A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d$.

Uniqueness. Suppose there exists two 7-tuples $(\mathbf{A}^0_1, \mathbf{B}^0_1, \mathbf{C}^0_1, \mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1, d_1)$ and $(\mathbf{A}^0_2, \mathbf{B}^0_2, \mathbf{C}^0_2, \mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_2, d_2)$ that satisfy the centering property specified in the lemma. Consider their differences $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d) = (\mathbf{A}^0_1 - \mathbf{A}^0_2, \mathbf{B}^0_1 - \mathbf{B}^0_2, \mathbf{C}^0_1 - \mathbf{C}^0_2, \mathbf{a}_1 - \mathbf{a}_2, \mathbf{b}_1 - \mathbf{b}_2, \mathbf{c}_1 - \mathbf{c}_2, d_1 - d_2)$. Clearly, we remain to show that $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ are zeros.

It is clear $\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0$ are doubly centered matrices and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are centered vectors. In addition, the following holds for all (i, j, k)

$$0 = A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d. \tag{5.21}$$

We first show $d = 0$. We can see this by summing over all i, j, k on both sides of Eq. (5.21),

$$0 = \sum_{ijk}[A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d] = n_1 n_2 n_3 d,$$

where the first inequality holds by the centering properties on $\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}$.

Next, we show that $\mathbf{a} = \mathbf{0}$. This can be done by summing over all $(j, k) \in [n_2] \times [n_3]$ on both sides of Eq. (5.21) and for any i,

$$\begin{aligned} 0 &= \sum_{jk} [A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d] \\ &= \sum_{jk} [A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k] \\ &= n_2 n_3 a_i, \end{aligned}$$

where we have used the result that $d = 0$ and the centering properties. Similarly, we can show $\mathbf{b} = \mathbf{0}$ and $\mathbf{c} = \mathbf{0}$.

Finally, we remain to show $\mathbf{A} = \mathbf{0}$. Again, fix any i, j and sum over all $k \in [n_3]$, we have

$$0 = \sum_{k} [A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d] = n_3 A^0_{ij},$$

where we have used the facts that $a_i = b_j = c_k = d = 0$. We can prove $\mathbf{B} = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$ using similar arguments. □

Lemma 5.8 essentially states that the representation of a pairwise interaction tensor is unique if one separate out these bias components. We can immmediately obtain Proposition 5.1 by condensing the unique representation scheme $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ for pairwise interaction tensors identified by Lemma 5.8. In particular, we construct $\mathbf{A}' \in S_A, \mathbf{B}' \in S_B, \mathbf{C}' \in S_C$ by setting $A'_{ij} = A^0_{ij} + a_i + d$, $B'_{jk} = B^0_{jk} + b_j$ and $C'_{ij} = C^0_{ki} + c_k$. By the centering property of $\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0$ and $\mathbf{a}, \mathbf{b}, \mathbf{c}$, it is clear that each column of $\mathbf{A}'$ sums up to a same value $(n_1 d)$ and each column of $\mathbf{B}', \mathbf{C}'$ sums up to zero. Hence $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ satisfy the constraints defined by S_A, S_B, S_C respectively. We can also easily show the uniqueness of $\mathbf{A}', \mathbf{B}', \mathbf{C}'$ under this constraints using the uniqueness of $(\mathbf{A}_0, \mathbf{B}_0, \mathbf{C}_0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$.

Proof. (Propostion 5.1) The existence follows immediately from Lemma 5.8. Specifically, we can set $A'_{ij} = A^0_{ij} + a_i + d$, $B'_{jk} =$

$B^0_{jk} + b_j$ and $C'_{ki} = C^0_{ki} + c_k$. We can easily verify that $\mathbf{A}' \in S_A, \mathbf{B}' \in S_B$ and $\mathbf{C}' \in S_C$.

Now we prove the uniqueness. Suppose that we have $\text{Pair}(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1) = \text{Pair}(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2)$, where $\mathbf{A}_1 \in S_A$, $\mathbf{A}_2 \in S_A$, $\mathbf{B}_1 \in S_B$, $\mathbf{B}_2 \in S_B$, $\mathbf{C}_1 \in S_C$ and $\mathbf{C}_2 \in S_C$. Denote $\mathbf{A} = \mathbf{A}_1 - \mathbf{A}_2$, $\mathbf{B} = \mathbf{B}_1 - \mathbf{B}_2$ and $\mathbf{C} = \mathbf{C}_1 - \mathbf{C}_2$, we remain to show that the differences $\mathbf{A}$,$\mathbf{B}$ and $\mathbf{C}$ are zero.

Note that $\mathbf{A} \in S_A, \mathbf{B} \in S_B, \mathbf{C} \in S_C$. We can construct 7-tuple $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ similarly to the proof of Lemma 5.8. We define the mean values of matrix $\mathbf{A}$ by $\sigma_A = \frac{1}{n_1 n_2}\mathbf{1}^T\mathbf{A}\mathbf{1}$ (note that the mean value of $\mathbf{B}$ and $\mathbf{C}$ is zero). We denote the mean vectors of rows of matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ by $\mathbf{a}_r = \frac{1}{n_2}\mathbf{A}\mathbf{1}$, $\mathbf{b}_r = \frac{1}{n_3}\mathbf{B}\mathbf{1}$ and $\mathbf{c}_r = \frac{1}{n_1}\mathbf{C}\mathbf{1}$.

Now, we construct the desired 7-tuple $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ by

$$A^0_{ij} = A_{ij} - a^r_i + \sigma_a, \quad B^0_{jk} = B_{jk} - b^r_k, \quad C^0_{ki} = C_{ki} - c^r_k,$$
$$a_i = a^r_i - \sigma_a,$$

where (i, j, k) ranges within $[n_1] \times [n_2] \times [n_3]$. In addition, we set $\mathbf{b} = \mathbf{b}_r, \mathbf{c} = \mathbf{c}_r$ and $d = \sigma_a$. We can verify that $A^0_{ij} + B^0_{jk} + C^0_{ki} + a_i + b_j + c_k + d = A_{ij} + B_{jk} + C_{ki} = 0$. By Lemma 5.8, it follows immediately that $(\mathbf{A}^0, \mathbf{B}^0, \mathbf{C}^0, \mathbf{a}, \mathbf{b}, \mathbf{c}, d)$ are zeros. Therefore, we have $\mathbf{A} = \mathbf{0}$, $\mathbf{B} = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$.

□

5.9 Details of recovery algorithm

Algorithm 8 Exact Recovery of Pairwise Interaction Tensor

```
procedure ExactRecover($\Omega = \{a_i b_i c_i\}_{i\in[m]}, \mathcal{P}_\Omega(\mathcal{T}) = \{T_{a_i b_i c_i}\}_{i\in[m]}, \tau, \delta, \epsilon$)
    $\mathbf{y} \leftarrow \mathbf{0}$
    for $k = 1, \dots, k_{\max}$ do
        $[\mathbf{X}, r_A] \leftarrow \text{shrink}_A(\mathcal{P}^*_{\Omega_A}(\mathbf{y}), \tau, r_A)$
        $[\mathbf{Y}, r_B] \leftarrow \text{shrink}_B(\mathcal{P}^*_{\Omega_B}(\mathbf{y}), \tau, r_B)$
        $[\mathbf{Z}, r_C] \leftarrow \text{shrink}_B(\mathcal{P}^*_{\Omega_C}(\mathbf{y}), \tau, r_C)$    ▷ $\text{shrink}_C$ is algorithmically identical to $\text{shrink}_B$.
        $\mathbf{e} \leftarrow \mathcal{P}_\Omega(\mathcal{T}) - \mathcal{P}_\Omega(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})$
        if $\|\mathbf{e}\|_F \,/\, \|\mathcal{P}_\Omega(\mathcal{T})\|_F \le \epsilon$ then
            break
        end if
        $\mathbf{y} \leftarrow \mathbf{y} + \delta\mathbf{e}$
    end for
end procedure
return $[n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}]$
```

Algorithm 9 Stable Recovery of Pairwise Interaction Tensor

```
procedure StableRecover($\Omega = \{a_i b_i c_i\}_{i\in[m]}, \mathcal{P}_\Omega(\hat{\mathcal{T}}) = \{\hat{T}_{a_i b_i c_i}\}_{i\in[m]}, \tau, \delta, \epsilon, \epsilon_1$)
    $\mathbf{y} \leftarrow \mathbf{0}$
    $s \leftarrow 0$
    for $k = 1, \dots, k_{\max}$ do
        $[\mathbf{X}, r_A] \leftarrow \text{shrink}_A(\mathcal{P}^*_{\Omega_A}(\mathbf{y}), \tau, r_A)$
        $[\mathbf{Y}, r_B] \leftarrow \text{shrink}_B(\mathcal{P}^*_{\Omega_B}(\mathbf{y}), \tau, r_B)$
        $[\mathbf{Z}, r_C] \leftarrow \text{shrink}_B(\mathcal{P}^*_{\Omega_C}(\mathbf{y}), \tau, r_C)$
        $\mathbf{e} \leftarrow \mathcal{P}_\Omega(\hat{\mathcal{T}}) - \mathcal{P}_\Omega(\text{Pair}(n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z})$
        if $\|\mathbf{e}\|_F \,/\, \left\|\mathcal{P}_\Omega(\hat{\mathcal{T}})\right\|_F \le \epsilon$ then
            break
        end if
        $\mathbf{y} \leftarrow \mathbf{y} + \delta\mathbf{e}$
        $s \leftarrow s - \delta\epsilon_1$
        $[\mathbf{y}, s] \leftarrow \mathcal{P}_{\mathcal{K}}(\mathbf{y}, s)$
    end for
end procedure
return $[n_3^{-1/2}\mathbf{X}, n_1^{-1/2}\mathbf{Y}, n_2^{-1/2}\mathbf{Z}]$
```

Algorithm 10 Shrinkage operator

```
procedure SHRINK$_B$($\hat{\mathbf{X}}, \tau, r$)
    $s \leftarrow r + 1$
    repeat
        $[\mathbf{U}, \Sigma, \mathbf{V}] \leftarrow$ svd(center($\hat{\mathbf{X}}$), $s$)  ▷ svd($\mathbf{M}, s$): return top $s$ singular vectors of $\mathbf{M}$
        $s \leftarrow s + 5$
    until $\sigma_{s-5} \leq \tau$
    $r \leftarrow \max\{j : \sigma_j > \tau\}$
    $\mathbf{X} \leftarrow \sum_{i=1}^{r} (\sigma_j - \tau)\mathbf{u}_j \mathbf{v}_j^*$
    return $[\mathbf{X}, r]$
end procedure
procedure SHRINK$_A$($\hat{\mathbf{X}}, \tau, r$)
    $[\mathbf{X}, r] \leftarrow$ shrink$_B$($\hat{\mathbf{X}}, \tau, r$)
    $\delta \leftarrow$ sum($\hat{\mathbf{X}}$)                ▷ sum($\hat{\mathbf{X}}$): elemetwise sum of matrix $\hat{\mathbf{X}}$
    $\gamma \leftarrow \frac{1}{n_1 n_2}(\{\delta - \tau\}_+ + \{\delta + \tau\}_-)$
    return $[\mathbf{X} + \gamma \mathbf{1}\mathbf{1}^T, r]$
end procedure
```

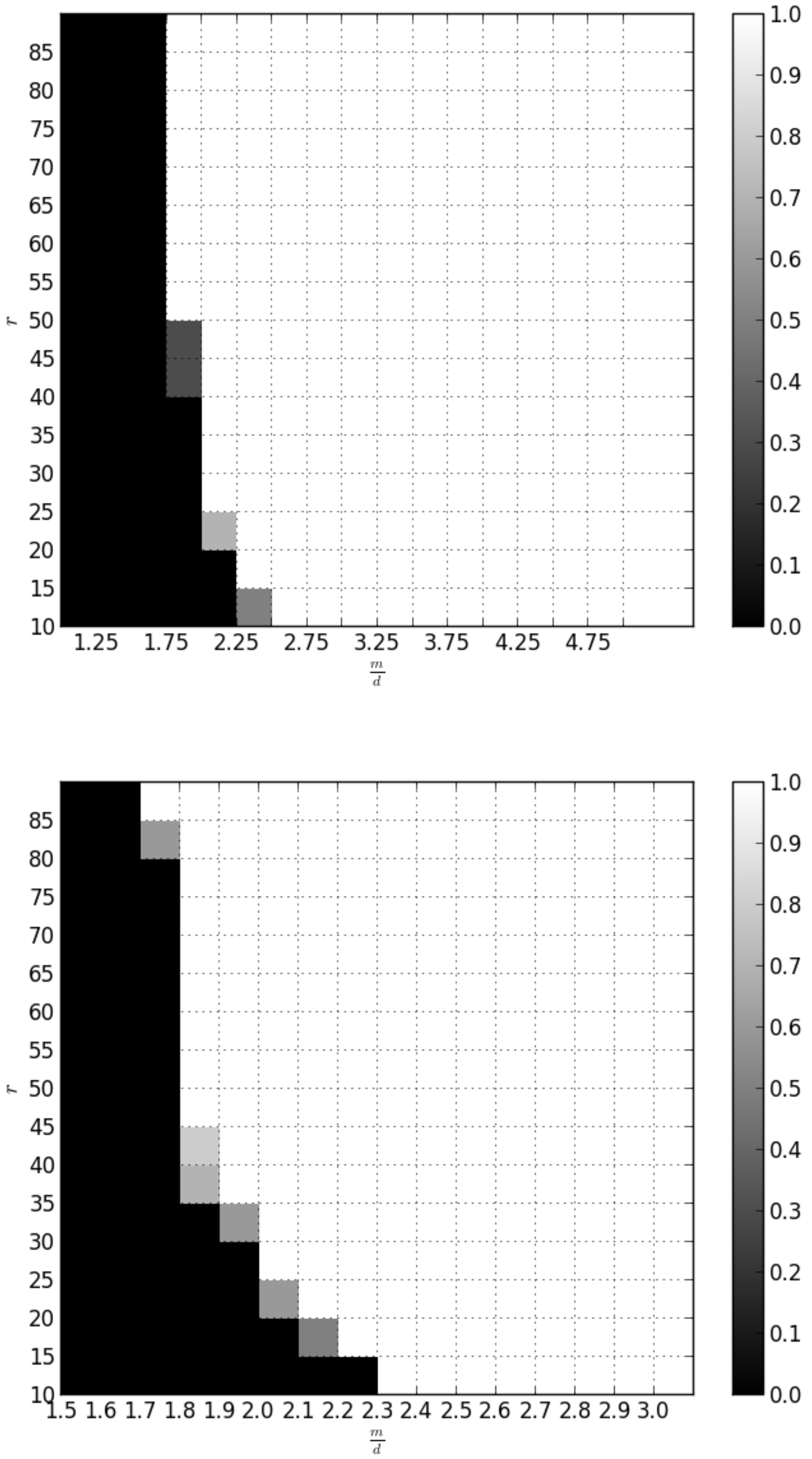
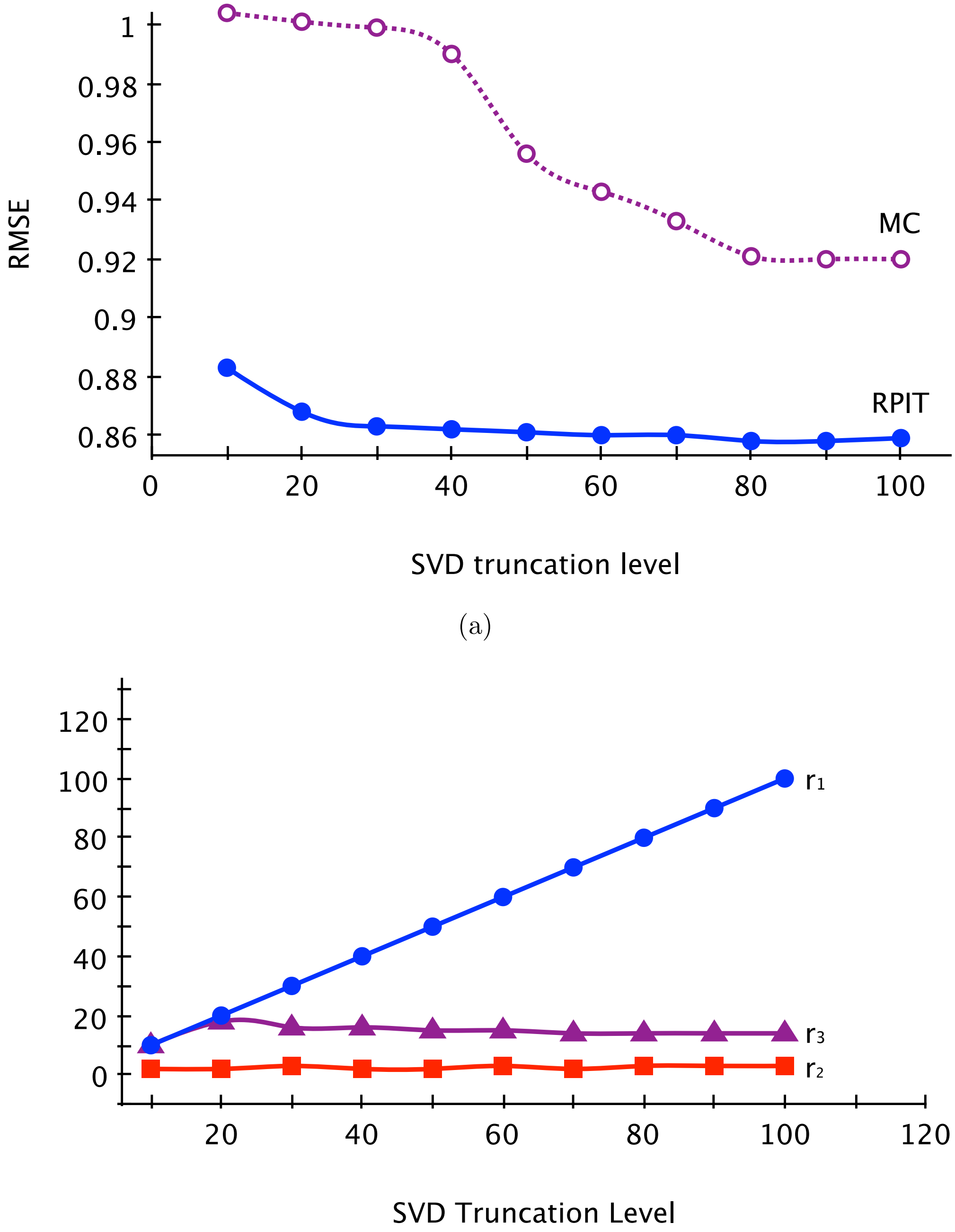


Figure 5.1: Phase transition with respect to rank and degree of freedom. Top: $m/d \in [1, 5]$. Bottom: $m/d \in [1.5, 3.0]$.



(a)

(b)

Figure 5.2: Empirical results on the Movielens dataset. (a) Comparison of RMSE with different truncation levels. MC: Matrix completion algorithm. RPIT: Recovery algorithm for pairwise interaction tensor. (b) Rank of recovered matrix $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. $r_1 = \text{rank}(\mathbf{X}), r_2 = \text{rank}(\mathbf{Y}), r_3 = \text{rank}(\mathbf{Z})$.

Chapter 6

Conclusion

In this thesis, we made theoretical and experimental contributions to several problems emerged from the fields of multi-armed bandit and tensor completion, which concern with learning in the presence of a limited number of samples. To each of these problems, we identified natural requirements on the number of samples and we designed practical algorithms with provable theoretical guarantees that are close to these requirements. Through our experiments, we demonstrated that our algorithms work well empirically and the experiment results agree well with our theories.

Specifically, in Chapter 2, we provided a series of results for the combinatorial pure exploration problem in the stochastic bandit setting, including general learning algorithms, upper bounds and a lower bound. In our analysis, we introduced a novel tool which may be of independent interest. Our upper and lower bounds showed that our algorithm achieve the optimal sample complexity (within logarithmic factors) in many cases. In addition, our lower bound partially settled a recent conjecture.

In Chapter 3, we applied combinatorial bandit methods to recommender systems. More specifically, we developed the linear combinatorial bandit model for the application of item list recommendation problem, in which the objective is to recommend lists of items over a sequence of rounds to a new user with insufficient historical records. Based on this model, we designed a learning algorithm that dynamically recommend diverse items that might be interested of a user. Further, our model generalizes the existing combinatorial bandit model and can be applied to other combinatorial bandit problems where arms are associated with feature vectors. We conducted experiments on a real-wold movie recommendation dataset and the results showed promising empirical performance of our algorithm.

Our learning algorithm for linear combinatorial bandits relies on a key algorithmic component called ridge regression, which is also widely applied in the fields of machine learning, data mining and statistics. In Chapter 4, we designed a fast relative-error approximation algorithm for ridge regression for the high-dimensional cases, i.e., the number of features p is larger than the number of samples n. We showed that our algorithm is the first algorithm for ridge regression that runs in $o(n^2p)$ time with provable relative-error approximation bound on the output vector. We also analyzed the risk inflation bound of our algorithm and generalized our technique to the multiple response ridge regression problem. Empirical results on both synthetic and real datasets demonstrated the efficiency and accuracy of our algorithm.

Finally, in Chapter 5, we studied the recovery problem of pairwise interaction tensors, a simplified low rank model for tensors, which has attracted considerable attention due to its simplicity and effectiveness. We designed an algorithm which, in the absence of noise, guarantees to exactly recover a pairwise interaction tensor from only $O(nr\log^2 n)$ random subsamples of entries, which is optimal within logarithmic factors. In addition, for noisy cases, we also developed a constrained convex program and prove the associated error bounds. Moreover, we proposed a simple and scalable algorithm for the related optimization problems. Our experiments demonstrated that our algorithm can be used to improve the accuracy of movie recommendation with time information.

List of Publications

[1] **Shouyuan Chen**, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. **Oral presentation** (To appear).

[2] Lijing Qin, **Shouyuan Chen**, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *SIAM International Conference on Data Mining (SDM)*, 2014. **Best Student Paper Award Runner-Up**.

[3] **Shouyuan Chen**, Yang Liu, Irwin King, Michael R. Lyu, and Shengyu Zhang. Fast relative-error approximation of ridge regression. Technical report, 2014.

[4] **Shouyuan Chen**, Michael R. Lyu, Irwin King, and Zenglin Xu. Exact and stable recovery of pairwise interaction tensors. In *Advances in Neural Information Processing Systems (NIPS)*, 2013. **Spotlight presentation**.

Bibliography

[1] Y. Abbasi-Yadkori, C. Szepesvári, and D. Tax. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274, 2008.

[3] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[4] R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *Information Theory, IEEE Transactions on*, 48(3):569–579, 2002.

[5] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards; part ii: Markoveian rewards. *Automatic Control, IEEE Transactions on*, 32(11):968–982, 1987.

[6] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *COLT*, 2010.

[7] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. *arXiv preprint arXiv:1105.4871*, 2011.

[8] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.

[9] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[10] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[11] H. Avron, C. Boutsidis, S. Toledo, and A. Zouzias. Efficient dimensionality reduction for canonical correlation analysis. In *ICML*, 2013.

[12] B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53. ACM, 2004.

[13] F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *COLT*, 2013.

[14] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[15] C. Beck and R. D'Andrea. Computational study and comparisons of lft reducibility methods. In *American Control Conference, 1998. Proceedings of the 1998*, volume 2, pages 1013–1017. IEEE, 1998.

[16] C. F. Beckmann and S. M. Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311, 2005.

[17] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[18] C. Berge. Two theorems in graph theory. *PNAS*, 1957.

[19] D. M. Berry. Bandit problems. 1985.

[20] M. Berry et al. Svdpackc (version 1.0) user's guide, university of tennessee tech. *Report (393-194, 1993 (Revised October 1996).*, 1993.

[21] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

[22] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *Neural Information Processing*, pages 324–331. Springer, 2012.

[23] J. Bourgain and J. Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. *Technical Report*, 2013.

[24] L. Breiman and J. H. Friedman. Predicting multivariate responses in multiple linear regression. *J R STAT SOC B*, 1997.

[25] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012.

[26] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412:1832–1852, 2010.

[27] S. Bubeck, N. Cesa-bianchi, S. M. Kakade, S. Mannor, N. Srebro, and R. C. Williamson. Towards minimax policies for online linear optimization with bandit feedback. In *COLT*, 2012.

[28] S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *ICML*, pages 258–265, 2013.

[29] R. Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12 (4):331–370, 2002.

[30] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[31] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

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

[33] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[34] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

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

[36] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

[37] F. Caro and J. Gallien. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53(2):276–292, 2007.

[38] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35 (3):283–319, 1970.

[39] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *JCSS*, 78(5):1404–1422, 2012.

[40] S. Chen, M. R. Lyu, I. King, and Z. Xu. Exact and stable recovery of pairwise interaction tensors. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[41] S. Chen, T. Lin, I. King, M. R. Lyu, and W. Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. To appear.

[42] S. Chen, L. Yang, I. King, M. R. Lyu, and S. Zhang. Fast relative-error approximation of ridge regression. Technical report, 2014.

[43] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *ICML*, pages 151–159, 2013.

[44] A. L. Chistov and D. Y. Grigor'ev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *Mathematical Foundations of Computer Science 1984*, pages 17–31. Springer, 1984.

[45] W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[46] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *STOC*, 2013.

[47] V. Dani, S. M. Kakade, and T. P. Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2007.

[48] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

[49] S. Dasgupta and A. Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *RS&A*, 2003.

[50] V. H. de la Peña, M. J. Klass, and T. L. Lai. Self-normalized processes: exponential inequalities, moment bounds and iterated logarithm laws. *Annals of probability*, pages 1902–1933, 2004.

[51] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[52] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[53] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *JMLR*, 2006.

[54] A. Evgeniou and M. Pontil. Multi-task feature learning. 2007.

[55] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[56] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference, 2001*, 2001.

[57] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003.

[58] M. Filipović and A. Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing*, pages 1–16, 2013.

[59] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *P IEEE*, 2005.

[60] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In *NIPS*. 2011.

[61] V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS*, 2012.

[62] Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012.

[63] Y. Gai, B. Krishnamachari, and R. Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–9. IEEE, 2010.

[64] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[65] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[66] A. Gopalan, S. Mannor, and Y. Mansour. Thompson sampling for complex online problems. In *ICML*, pages 100–108, 2014.

[67] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *Information Theory, IEEE Transactions on*, 57(3):1548–1566, 2011.

[68] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Physical review letters*, 105(15):150401, 2010.

[69] I. Guyon. Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, 2003.

[70] R. A. Harshman. Foundations of the parafac procedure: models and conditions for an" explanatory" multimodal factor analysis. 1970.

[71] J. Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[72] E. Hazan and S. Kale. Beyond convexity: Online submodular minimization. In *Advances in Neural Information Processing Systems*, pages 700–708, 2009.

[73] C. Hillar and L.-H. Lim. Most tensor problems are np hard. *JACM*, 2013.

[74] F. L. Hitchcock. *The expression of a tensor or a polyadic as a sum of products*. sn., 1927.

[75] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[76] P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, 2010.

[77] K. Jamieson and R. Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2014.

[78] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'UCB: An optimal exploration algorithm for multi-armed bandits. *COLT*, 2014.

[79] S. Kale, L. Reyzin, and R. E. Schapire. Non-stochastic bandit slate problems. In *NIPS*, 2010.

[80] S. Kalyanakrishnan and P. Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, pages 511–518, 2010.

[81] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *ICML*, pages 655–662, 2012.

[82] E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *COLT*, 2013.

[83] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[84] H. A. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of chemometrics*, 14 (3):105–122, 2000.

[85] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[86] Y. Koren. Collaborative filtering with temporal dynamics. In *SIGKDD*, 2009.

[87] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer, 2011.

[88] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

[89] A. Krishnamurthy and A. Singh. Low-rank matrix and tensor completion via adaptive sampling. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Process-*

ing Systems 26, pages 836–844. Curran Associates, Inc., 2013.

[90] B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *UAI*, 2014.

[91] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6 (1):4–22, 1985.

[92] R. M. Larsen. Propack-software for large and sparse svd calculations. *Available online.*, 2004.

[93] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[94] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

[95] T. Lin, B. Abrahao, R. Kleinberg, J. Lui, and W. Chen. Combinatorial partial monitoring game with linear feedback and its application. In *ICML*, 2014.

[96] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[97] H. Liu, K. Liu, and Q. Zhao. Logarithmic weak regret of non-bayesian restless multi-armed bandit. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1968–1971, 2011.

[98] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *ICCV*, 2009.

[99] K. Liu and Q. Zhao. Adaptive shortest-path routing under unknown and stochastically varying link states. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on*, pages 232–237. IEEE, 2012.

[100] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[101] M. W. Mahoney, M. Maggioni, and P. Drineas. Tensor-cur decompositions for tensor-based data. *SIAM Journal on Matrix Analysis and Applications*, 30(3):957–987, 2008.

[102] S. Mannor and J. N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648, 2004.

[103] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.

[104] H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Learning theory*, pages 109–123. Springer, 2004.

[105] M. Mesbahi and G. P. Papavassilopoulos. On the rank minimization problem over a positive semidefinite linear matrix inequality. *Automatic Control, IEEE Transactions on*, 42(2):239–243, 1997.

[106] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

[107] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[108] MOVIELENS-DATA. MovieLens dataset. In *http://movielens.org.*

[109] J. Nelson and H. L. Nguyen. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *FOCS*, 2013.

[110] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[111] G. Neu, A. György, and C. Szepesvári. The online loop-free stochastic shortest-path problem. In *COLT*, pages 231–243, 2010.

[112] J. G. Oxley. *Matroid theory.* Oxford university press, 2006.

[113] V. H. Peña, H. Víctor, T. L. Lai, and Q.-M. Shao. *Self-Normalized Processes: Limit Theory and Statistical Applications.* Springer, 2009.

[114] D. Pollard. Asymptopia. *Manuscript, Yale University, Dept. of Statist., New Haven, Connecticut*, 2000.

[115] I. Porteous, E. Bart, and M. Welling. Multi-hdp: A non-parametric bayesian model for tensor factorization. In *AAAI*, 2008.

[116] L. Qin and X. Zhu. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the Twenty-third international joint conference on Artificial Intelligence*, pages 2698–2704. AAAI Press, 2013.

[117] L. Qin, S. Chen, and X. Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *SIAM International Conference on Data Mining (SDM)*, 2014.

[118] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM, 2008.

[119] B. Recht. A simpler approach to matrix completion. *JMLR*, 2011.

[120] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[121] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *ICDM*, 2010.

[122] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *SIGKDD*, 2009.

[123] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[124] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings*

of the 22nd international conference on Machine learning, pages 713–719. ACM, 2005.

[125] O. Rivasplata. Subgaussian random variables: An expository note. 2012.

[126] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[127] S. M. Ross. *Stochastic processes*, volume 2. John Wiley & Sons New York, 1996.

[128] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *NIPS*, 2008.

[129] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, 2006.

[130] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[131] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998.

[132] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[133] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276–280, 1986.

[134] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML*, 2005.

[135] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. Tfmap: Optimizing map for top-n context-aware recommendation. In *SIGIR*, 2012.

[136] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, 94(3):303–351, 2014.

[137] A. Singer. A remark on global positioning from local distances. *Proceedings of the National Academy of Sciences*, 105(28):9507–9511, 2008.

[138] A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

[139] H. Sorensen, D. Jones, C. Burrus, and M. Heideman. On computing the discrete hartley transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1985.

[140] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4):133–145, 2002.

[141] N. Srebro. *Learning with matrix factorizations*. PhD thesis, Citeseer, 2004.

[142] N. Srebro, J. D. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. *NIPS*, 2005.

[143] M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. Technical report, DTIC Document, 2007.

[144] T. Tao. *Topics in random matrix theory*. 2012.

[145] G. Tomasi and R. Bro. Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75(2): 163–180, 2005.

[146] R. Tomioka and T. Suzuki. Convex tensor decomposition via structured schatten norm regularization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1331–1339. Curran Associates, Inc., 2013.

[147] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[148] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. *NIPS*, 2011.

[149] J. A. Tropp. Improved analysis of the subsampled randomized hadamard transform. *AADA*, 2011.

[150] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[151] J. Weston, C. Wang, R. Weiss, and A. Berenzweig. Latent collaborative retrieval. ICML, 2012.

[152] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.

[153] Z. Xu, F. Yan, et al. Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. *arXiv preprint arXiv:1108.6296*, 2011.

[154] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *Proceedings of the 12th international conference on extending database technology: Advances in database technology*, pages 368–378. ACM, 2009.

[155] Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, pages 2483–2491, 2011.

[156] Y. Zhou, X. Chen, and J. Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In *ICML*, 2014.