2021

On Connections Between Machine Learning And Information Elicitation, Choice Modeling, And Theoretical Computer Science

Arpit Agarwal
University of Pennsylvania

On Connections Between Machine Learning And Information Elicitation, Choice Modeling, And Theoretical Computer Science

Abstract

Machine learning, which has its origins at the intersection of computer science and statistics, is now a rapidly growing area of research that is being integrated into almost every discipline in science and business such as economics, marketing and information retrieval. As a consequence of this integration, it is necessary to understand how machine learning interacts with these disciplines and to understand fundamental questions that arise at the resulting interfaces. The goal of my thesis research is to study these interdisciplinary questions at the interface of machine learning and other disciplines including mechanism design/information elicitation, preference/choice modeling, and theoretical computer science.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

Shivani Agarwal

Keywords

Choice Modeling, Information Elicitation, Machine Learning, Ranking, Theoretical Computer Science

Subject Categories

Computer Sciences

ON CONNECTIONS BETWEEN MACHINE LEARNING AND INFORMATION ELICITATION, CHOICE MODELING, AND THEORETICAL COMPUTER SCIENCE

Arpit Agarwal

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2021

Supervisor of Dissertation

Shivani Agarwal, Rachleff Family Associate Professor of Computer and Information Science

Graduate Group Chairperson

Mayur Naik, Professor of Computer and Information Science

Dissertation Committee

Sanjeev Khanna, Henry Salvatori Professor of Computer and Information Science
Rakesh Vohra, George A. Weiss and Lydia Bravo Weiss University Professor
Hamed Hassani, Assistant Professor of Electrical and Systems Engineering
David Parkes, George F. Colony Professor of Computer Science, John A. Paulson School of Engineering and Applied Sciences, Harvard University

ON CONNECTIONS BETWEEN MACHINE LEARNING AND INFORMATION ELICITATION, CHOICE MODELING, AND THEORETICAL COMPUTER SCIENCE

Dedicated to my grandmother

ACKNOWLEDGEMENT

Firstly, I would like to warmly thank my advisor, Shivani Agarwal, for her patience and guidance during the past few years. When I first met you during your Probability and Statistics class at IISc, I did not know what research really means. It is through your continued inspiration that I have been able to learn the meaning of research. I am really grateful for this inspiration and guidance that has had a profound impact on my life. Also, thank you for lending me an ear whenever I needed one and for always being there for me.

Secondly, I would like to thank Sanjeev Khanna teaching me how to have fun while doing research. You always manage to make me excited about something whenever I talk to you. I would also like to thank David Parkes for hosting me at Harvard for a semester and always finding time for me during your busy schedule. I would also like to thank all members of my thesis committee– Shivani Agarwal, Hamed Hassani, Sanjeev Khanna, David Parkes and Rakesh Vohra– for giving valuable feedback on my thesis.

This thesis would have not been possible without my excellent collaborators: Shivani Agarwal, Sepehr Assadi, Ashwinkumar BV, Rafael Frongillo, Sanjeev Khanna, Debmalya Mandal, Harikrishna Narasimhan, David C. Parkes, Nisarg Shah, and Victor Shnayder. I would also like to thank my friends– Prathamesh, Dushyant, Arun, Hari, Harish, Akhil, Manish, Nand, Vineet– who have always been by my side in tough times.

Lastly I would like to thank my family, especially my parents, for all the sacrifices they have made for this to be possible. I would like to thank my mother for being there when no one was, and my father for loving me unconditionally.

ABSTRACT

ON CONNECTIONS BETWEEN MACHINE LEARNING AND INFORMATION ELICITATION, CHOICE MODELING, AND THEORETICAL COMPUTER SCIENCE

Arpit Agarwal

Shivani Agarwal

Machine learning, which has its origins at the intersection of computer science and statistics, is now a rapidly growing area of research that is being integrated into almost every discipline in science and business such as economics, marketing and information retrieval. As a consequence of this integration, it is necessary to understand how machine learning interacts with these disciplines and to understand fundamental questions that arise at the resulting interfaces. The goal of my thesis research is to study these interdisciplinary questions at the interface of machine learning and other disciplines including mechanism design/information elicitation, preference/choice modeling, and theoretical computer science.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF PUBLICATIONS BASED ON THIS THESIS

1. Agarwal, A., Johnson, N., Agarwal, S.,

   Choice Bandits.

   In Neural Information Processing Systems (**NeurIPS**), 2020.

2. Agarwal, A., Mandal, D., Parkes, D., and Shah, N.,

   Peer Prediction with Heterogeneous Users.

   In ACM Transactions on Economics and Computation (**TEAC**), 2020.

   A shorter version appeared in ACM Conference on Economics and Computation (**EC**), 2017.

   Note: This work is a joint contribution of this thesis and Mandal, D.'s thesis.

3. Agarwal, A., Assadi, S., and Khanna, S.,

   Stochastic Submodular Covering with Limited Adaptivity.

   In ACM-SIAM Symposium on Discrete Algorithms (**SODA**), 2019

4. Agarwal, A., Patil, P., and Agarwal, S.,

   Accelerated Spectral Ranking.

   In International Conference on Machine Learning (**ICML**), 2018.

5. Agarwal, A., Agarwal, S., Assadi, S., and Khanna, S.,

   Learning with Limited Rounds of Adaptivity: Coin Tossing, Multi-Armed Bandits, and Ranking from Pairwise Comparisons.

   In Conference on Learning Theory (**COLT**), 2017.

   Note: A part of this work is a contribution of Assadi, S.'s thesis.

6. Agarwal, A. and Agarwal, S.,

   On Consistent Surrogate Risk Minimization and Property Elicitation.

   In Conference on Learning Theory (**COLT**), 2015.

Chapter 1

Introduction

Machine learning (ML), which has its origins at the intersection of computer science and statistics, has recently seen remarkable success in a wide range of applications including image recognition, information retrieval, recommendation systems, medical diagnosis, and many more. The empirical success in these wide ranging applications has naturally led to the integration of ML in many other disciplines of science and business.

On one hand, traditional approaches in these disciplines are being augmented with machine learning methods so as to improve these approaches along several dimensions. For example, traditional approaches in econometrics like A/B testing are being augmented/replaced with more sample efficient algorithms from online/active learning (Athey and Imbens, 2019), mechanism design algorithms are using machine learning techniques in order to relax assumptions about the underlying data distribution (Agarwal et al., 2017b), and traditional combinatorial algorithms are using ML predictions so as to improve their performance (Purohit et al., 2018). On the other hand, ideas/concepts from other disciplines are also making their way into machine learning and are proving to be of importance to the science of machine learning. For example, ideas from probability forecasting and computational economics literature are helping to better understand the design of loss functions in ML (Agarwal and Agarwal, 2015; Liu and Guo, 2020; Liu and Helmbold, 2020), probabilistic models for human decision-making studied in econometrics are making their way into machine learning and finding application in various web applications (Ie et al., 2019), ideas about resource-constrained computing from theoretical computer science are making their way to machine learning in order to enable efficient parallel/distributed learning (Konevcny et al., 2016; Agarwal et al., 2017a).

While remarkable progress has been made in the science of machine learning, its integration with many disciplines in science and business is still relatively new. Hence, there are still a lot of gaps in our end-to-end understanding of its interaction with these other disciplines. Therefore, it is important to study the fundamental questions resulting from such interactions in order to fill these gaps in our understanding.

The goal of this thesis is to study *interdisciplinary* questions that arise at the interface of machine learning and other disciplines including mechanism design/information elicitation, preference/choice modeling, and theoretical computer science. A common theme in this thesis is the use of mathematical formalism and theoretical analysis in order to first understand the powers and limitations of current approaches for these problems, and then guide the design of improved and principled solutions. Through this interdisciplinary study, this thesis has contributed towards the creation of two-way knowledge bridges between machine learning and other fields including information elicitation/mechanism design, choice/preference elicitation, theoretical computer science, leading to the design of principled solutions for common problems. I will describe below the three broad interfaces that I have explored in my research and describe the contributions in each of these in more detail. The following will also serve as a roadmap for the rest of the thesis.

1.1 Interface Between Machine Learning and Information Elicitation

Information elicitation, which is studied in economics and statistics, is the design of mechanisms that incentivize *strategic* humans to *truthfully* exchange their beliefs, for example prediction market mechanisms for eliciting beliefs about (uncertain) future events. My research at the interface of information elicitation and machine learning has led to new understanding about how viewing supervised learning algorithms as information elicitation mechanisms can help in the design of new loss functions for learning (Agarwal and Agarwal, 2015); and how machine learning can help in designing better mechanisms for information elicitation in the absence of ground truth (Agarwal et al., 2017a).

Chapter 2– Calibrated surrogate losses and proper scoring rules. Minimization of *calibrated surrogate loss* functions, such as logistic and hinge loss, is a widely used framework in *consistent* supervised learning (Bartlett et al., 2006; Tewari and Bartlett, 2007); scoring agents using *proper scoring rules*, such as log and Huber scoring rules, is a widely used framework in *truthful* information elicitation (Savage, 1971; Gneiting and Raftery, 2007). It is well-known that there exists a correspondence between calibrated surrogate losses and proper scoring rules: certain surrogate losses such as the logisitc or cross-entropy loss, can be viewed as proper scoring rules for eliciting the *complete conditional label distribution* given an instance (Buja et al., 2005; Reid and Williamson, 2010). However, this correspondence was previously understood to hold for a fairly limited class of surrogates, as not all surrogates can be viewed as eliciting the complete underlying label distribution.

In this thesis we show a much stronger correspondence between calibrated surrogates and proper scoring rules: *a large class of calibrated surrogate losses in supervised learning* can essentially be viewed as *proper scoring rules* for eliciting or estimating *certain properties* of the underlying conditional label distribution that are sufficient to construct an optimal classifier; and conversely, a large class of proper scoring rules can be viewed as calibrated surrogates for supervised learning problems. For example, we show that several surrogate loss functions for the problem of subset ranking, such as the least-squares surrogates of Ramaswamy et al. (2013), can essentially be viewed as eliciting linear properties of the underlying label distribution. This connection also gives a way to design efficient calibrated surrogates for supervised learning using the theory of proper scoring rules.

The materials in this chapter are based on a joint paper with Shivani Agarwal (Agarwal and Agarwal, 2015) in COLT'15.

Chapter 3– Information elicitation in the absence of ground truth. Typically, a scoring rule is designed to take as input a report from an agent and a ground truth sample. However, in many applications of information elicitation, such as the ones involving crowdsourcing for machine learning, there is no ground truth sample available. *Peer prediction* is the general framework for designing *truthful* mechanisms in this setting that score an agent by using reports of randomly chosen peer agents as the proxy for a ground truth sample. The problems in designing practical peer prediction mechanisms, however, have been the presence of uninformative equilibria where the agents can just 'agree to agree' and maximize their scores (Jurca and Faltings, 2005); and the fact that these mechanisms are only truthful when all agents have *homogeneous* beliefs (Radanovic and Faltings, 2015b).

In this thesis *we design the first peer prediction mechanism that has truthfulness guarantees for heterogeneous agents* and also avoids the problem of uninformative equilibria. We use machine learning techniques to cluster the users based on similarity of reports and extend our mechanism from Shnayder et al. (2016a) to work with these clusters of 'almost' homogeneous users. This forms a closed loop between machine learning and information elicitation, where information elicitation mechanisms can be used to collect truthful data for machine learning; and machine learning can be used to learn the best mechanism out of all possible mechanisms for information elicitation.

The materials discussed here are based on a joint paper with Debmalya Mandal, David Parkes, and Nisarg Shah (Agarwal et al., 2017b) in EC'17.

1.2 Interface between Machine Learning and Choice Modeling

Discrete choice modeling, which is studied in a variety of fields including economics and transportation, is concerned with the design of models of how humans make choices given a set of alternatives. The emergence of online services in domains including entertainment and shopping, that use machine learning to recommend alternatives to users, has presented unique challenges at the interface of discrete choice modeling and machine learning. This thesis addresses some of these challenges by developing fast and statistically efficient algorithms for estimating the parameters of the multinomial logit (MNL) choice model (Agarwal et al., 2018), and developing a multi-armed bandit framework for identifying (recommending) ‘best’ (‘good’) items with respect to a (unknown) discrete choice model (Agarwal et al., 2019b).

Chapter 4– Learning multinomial logit (MNL) model from choices. We study the problem of learning the parameters of the multinomial logit (MNL) choice model, which is one of the most widely studied models in discrete choice, using (offline) data about choices made by a user when presented with different alternatives. We develop a spectral algorithm for learning this model, which is orders of magnitude *faster* in computation time than existing algorithms (Negahban et al., 2017; Maystre and Grossglauser, 2015), can be implemented in a distributed setting, and is also *statistically more efficient* than previous algorithms (Negahban et al., 2017).

The materials in this chapter are based on a joint paper with Shivani Agarwal and Prathamesh Patil (Agarwal et al., 2018) in ICML’18.

Chapter 5– Multi-armed bandits and discrete choice models. How can humans discover *good* items which they have never interacted with in the past? In other words, how can we balance the ‘exploitation’ of items which we already know that the user has a ‘decent’ preference for, with ‘exploration’ of more items in order to learn more about user preference? The framework of *multi-armed bandits* seeks to balance this ‘exploitation’ and ‘exploration’ trade-off by minimizing an appropriate notion of *regret* over a sequence of interactions. In this thesis we develop a new framework, which we term as *choice bandits*, where a learner offers a choice set of items to a user in each round of interaction and the user chooses an item from this set according to an underlying (unknown) choice model. The regret is defined in terms of the overall quality of the choice sets with respect to a ‘best’ item in the choice model. We develop an efficient algorithm for this problem which has a *sublinear regret* for a wide variety of choice models including random utility models. Our study also *opens up several questions at the interface of multi-armed bandits and discrete choice models*, for example, designing low-regret algorithms for a broader class of choice models such as mixture of MNLs.

The materials in this chapter are based on a joint paper with Shivani Agarwal and Nicholas Johnson (Agarwal et al., 2020). A short version of this paper appeared in NeurIPS'20 and a longer version is in preparation for submission to a journal.

1.3 Interface Between Machine Learning and Theoretical Computer Science

In recent years there have been many avenues for exchange of ideas between machine learning and theoretical computer science. One such avenue is the design of parallel algorithms, which has been an important research direction in theoretical computer science, but is now becoming increasingly popular in machine learning. This popularity is driven by the fact many active/adaptive machine learning algorithms, such as those used in ad placement, are highly adaptive (sequential) in their ability to process data even though they can collect data in parallel from different users. I have contributed to the design of algorithms that have low adaptivity for important problems in both machine learning and theoretical computer science including best arm identification in multi-armed bandits (Agarwal et al., 2017b) and stochastic submodular covering (Agarwal et al., 2019a).

Chapter 6– Multi-armed bandits with limited adaptivity. Best arm identification is a widely studied problem in multi-armed bandits where the goal is to find an arm with the highest expected reward among a finite set of stochastic arms by repeatedly pulling (sampling reward from) these arms. Most algorithms for this problem are highly adaptive, i.e. the algorithm only pulls an arm after observing the results of all the previous pulls. In this thesis we study algorithms that solve this problem in a limited number of adaptive rounds, where in each round the algorithm pulls arms in parallel. We design an algorithm that *improves more than exponentially over previous algorithms in terms of rounds of adaptivity*, while requiring the same number of pulls as the previous best algorithm (Even-Dar et al., 2006).

The materials in this chapter are based on a joint paper with Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna (Agarwal et al., 2017a) in COLT'17.

Chapter 7– Stochastic submodular cover with limited adaptivity. Submodular optimization is well-studied in combinatorial optimization and theoretical computer science, but has also gained a lot of attention in machine learning recently, due to its applications in diverse data collection, data summarization, viral marketing etc. An important problem in this area is that of stochastic submodular covering where there is a submodular set function that takes different values depending upon a stochastic environment, and the goal is to adaptively probe the function value on different sets until a desired function value is reached (Golovin and Krause, 2010). In this thesis we study algorithms that probe sets in parallel and only use a few adaptive rounds. We show *tight bounds* on the number of probes required to solve the problem given a fixed number of rounds of adaptivity.

The materials in this chapter are based on a joint paper with Sepehr Assadi and Sanjeev Khanna (Agarwal et al., 2019a) in SODA'19.

1.4 Some Comments on Additional Connections

In this section we will outline broader themes underlying some of the problems studied in this thesis and discuss connections with existing literature.

- **Heterogeneity:** The two interfaces discussed in Section 1.1 and Section 1.2 are concerned with eliciting/aggregating/learning the beliefs/preferences of humans. It is well-understood that humans are heterogeneous in their beliefs/preferences, and hence, taking into account this heterogeneity is an important direction of research at these interfaces. There is already substantial literature on heterogeneity at the interface between machine learning and information elicitation, for example Chapter 3 in this thesis studies mechanisms for elicitation of heterogeneous beliefs in the absence of ground truth using machine learning techniques; Simpson et al. (2013) study the role of heterogeneity in aggregating human labels for machine learning tasks; and Zhang et al. (2015) study the role of multi-armed bandit algorithms for allocating crowdsourcing tasks to humans that have a varying level of accuracy on different tasks. The interface of machine learning and choice modeling also contains a fast growing literature on the study of choice models that take into account heterogeneity, for example Awasthi et al. (2014) study the learnability of a mixtures of two Mallows model; Zhao and Xia (2019); Liu et al. (2019); Chierichetti et al. (2018); Oh and Shah (2014) study the learnability of a mixture of multinomial logit (MMNL) models. In the future we expect to see more work on incorporating heterogeneity for many problems at these interfaces.

- **Parallelism/Adaptivity:** As discussed in Section 1.3, the design of parallel/less adaptive algorithms is an active direction of research in machine learning and spans across many areas. Apart from the two areas discussed in Section 1.3, there are several other areas such as regret minimization in multi-armed bandits, ranking from pairwise comparisons, clustering etc., where adaptivity has been studied. Perchet et al. (2015b) study the tradeoff between adaptivity and regret in the regret minimization setting for two-armed bandits where the goal is to minimize the regret of an algorithm that pulls arms in batches (parallely). Gao et al. (2019) further extend the results of Perchet et al. (2015b) to multiple arms. Ruan et al. (2021); Esfandiari et al. (2021) study the tradeoff between regret and adaptivity for linear contextual bandits. Braverman et al. (2019); Cohen-Addad et al. (2020) study the tradeoff between adaptivity and sample complexity for the problem of ranking from pairwise comparisons under a *noisy comparison* model. Cohen-Addad et al. (2021) study the design of parallel algorithms for the problem of correlation clustering. In the future we expect to have more literature on the tradeoffs that arise due to paralleism/adaptivity for many other problems in machine learning.

Starting from the next chapter, we delve into details and present our results for each of these problems along with formal proofs of correctness. Each chapter is designed to be self-contained and can be read independently of the other chapters.

Chapter 2

Calibrated Surrogate Losses and Proper Scoring Rules

In this chapter we will start our discussion at the interface between machine learning and information elicitation. We will show a close relation between surrogate risk minimization which is a popular framework for supervised learning, and property elicitation which is a widely studied area in probability forecasting, statistics and economics.

2.1 Introduction

2.1.1 Background and Motivation

Surrogate risk minimization is one of the most popular algorithmic frameworks for supervised learning problems such as 0-1 (binary) classification, subset ranking, multilabel classification and others; and has been well-studied in the machine learning and learning theory community in recent years (Bartlett et al., 2006; Zhang, 2004a,b; Tewari and Bartlett, 2007; Steinwart, 2007; Cossock and Zhang, 2008; Xia et al., 2008; Duchi et al., 2010; Buffoni et al., 2011; Ravikumar et al., 2011; Calauzènes et al., 2012; Lan et al., 2012; Ramaswamy and Agarwal, 2012; Ramaswamy et al., 2013). Under this framework, given a target loss or performance measure of interest such as the 0-1 binary classification loss, the goal is to design a convex surrogate loss such as the hinge loss which can be efficiently optimized in a learning algorithm. It is also desirable that the surrogate is *calibrated*, i.e. minimization of the surrogate loss should *effectively* result in the minimization of the target loss in the limit of infinite samples.

Property elicitation is a widely used framework in information elicitation, and has been well-studied in the probability forecasting literature and has recently received renewed interest in the machine learning, statistics, and economics communities (Savage, 1971; Schervish, 1989; Gneiting and Raftery, 2007; Lambert et al., 2008; Lambert and Shoham, 2009; Vernet et al., 2011; Abernethy and Frongillo, 2012; Steinwart et al., 2014). Under this framework, given a target property/function of an unknown distribution (e.g. mean) the goal is to design a scoring rule (e.g. Brier score) which can be used to score agents' reports against samples from the underlying distribution. It is desirable that the scoring rule is *proper*, i.e. the correct value of the property is a minimizer of the scoring rule in the limit of infinite samples.

It is well-known that there exist similarities between several surrogate losses used for binary classification and scoring rules used for eliciting the Bernoulli distribution (Buja et al., 2005; Reid and Williamson, 2010; Menon and Williamson, 2016; Narasimhan and Agarwal, 2013). For example, Buja et al. (2005); Reid and Williamson (2010) showed that any proper scoring rule for eliciting the Bernoulli distributions, such as the log scoring rule, can be composed with an appropriate *link function* to construct a calibrated surrogate for binary classification such as the logistic loss. In other words, certain calibrated surrogates for binary classification can essentially be viewed as eliciting the Bernoulli conditional label distribution. Williamson et al. (2016) extended this correspondence beyond the binary case and showed that several surrogate losses for multiclass classification effectively elicit the multinomial conditional label distribution.

However, this correspondence was previously understood to hold for a fairly limited class of surrogates as not all surrogates can be viewed as eliciting the *complete conditional label distribution*. This excludes many surrogates for binary/multiclass classification such as the hinge (Zhang, 2004a); and almost all surrogates for problems with large label spaces (e.g. subset ranking) where it is highly inefficient to elicit the complete conditional label distribution. Does this mean that such surrogates are completely unrelated to proper scoring rules or is there a correspondence? Are these surrogates eliciting some other succinct property of the conditional label distribution rather than eliciting the entire distribution? In this chapter we aim to understand these questions and seek to establish a stronger connection between calibrated surrogates and proper scoring rules.

2.1.2 Our Contributions

In this chapter we define the notion of a *calibrated property* for a target loss function, such that the optimal prediction under the target loss can be constructed using this property. We show that given any target loss function, any strictly proper scoring rule for eliciting this calibrated property results in a calibrated surrogate loss. Conversely, we show that any calibrated surrogate can be used as a proper scoring rule for eliciting a calibrated property. This implies that a large class of calibrated surrogate losses in supervised learning can essentially be viewed as proper scoring rules for eliciting calibrated properties of the underlying conditional label distribution, and a large class of proper scoring rules can essentially be viewed as calibrated surrogates for certain target loss functions.

We use this framework to study the design of convex calibrated surrogates using proper scoring rules for linear and nonlinear properties. We show how the standardization functions studied by Buffoni et al. (2011) for subset ranking losses, as well as the general least-squares type surrogates studied by Ramaswamy et al. (2013), effectively amount to estimating linear properties of the distribution. We then show how using nonlinear properties can allow for the design of lower-dimensional convex calibrated surrogates. One offshoot of our work is a new framework for studying low-noise conditions; we show that eliciting a vector of quantiles allows one to obtain interval estimates of the label probabilities, based on which one can construct calibrated surrogates under any such condition where such a coarse probability estimate suffices to find an optimal classifier.

2.1.3 Notation

For $n \in \mathbb{Z}_+$, denote $[n] = \{1, \ldots, n\}$ and $\Delta_n = \{\mathbf{p} \in \mathbb{R}^n_+ : \sum_{i=1}^n p_i = 1\}$. Denote by $\mathcal{S}_n$ the set of permutations on n objects. For $\mathbf{u} \in \mathbb{R}^n$, denote $\mathrm{argsort}(\mathbf{u}) = \big\{\sigma \in \mathcal{S}_n : u_i > u_j \implies \sigma(i) < \sigma(j)\,,\ \forall i, j \in [n]\big\}$. For a set $A \subseteq \mathbb{R}^n$, denote by $\mathrm{relint}(A)$ the relative interior of A, by $\mathrm{bndry}(A)$ the boundary of A, and by $\dim(A)$ the dimension of the affine extension of A. For a matrix $\mathbf{L} \in \mathbb{R}^{n \times k}$, denote by $\mathrm{col}(\mathbf{L})$ the column-space of $\mathbf{L}$, and by $\mathrm{affdim}(\mathbf{L})$ the affine dimension of the set of columns of $\mathbf{L}$. For a strictly convex function $\phi : \mathbb{R}^n \to \mathbb{R}$, denote by B_ϕ the Bregman divergence with respect to ϕ, defined as $B_\phi(\mathbf{u}_1, \mathbf{u}_2) = \phi(\mathbf{u}_1) - \phi(\mathbf{u}_2) - \partial\phi_{\mathbf{u}_2}^\top(\mathbf{u}_1 - \mathbf{u}_2)$ where $\partial\phi_{\mathbf{u}_2}$ denotes a subderivative of ϕ at $\mathbf{u}_2$.

2.1.4 Organization

In Section 2.2 we set up some preliminaries related to surrogate risk minimization and property elicitation. In Section 2.3 we define the notion of calibrated properties and give our main result. In Section 2.4 we study the design of calibrated surrogates via linear properties and in Section 2.5 the design of calibrated surrogates via non-linear properties.

2.2 Preliminaries

We set up some preliminaries related to surrogate risk minimization in Section 2.2.1 and property elicitation in Section 2.2.2; the rest of the chapter will then connect these two themes.

2.2.1 Surrogate Risk Minimization and Calibrated Surrogates

We consider supervised learning problems with instance space $\mathcal{X}$, finite label space $\mathcal{Y} = [n]$, and finite prediction space $\widehat{\mathcal{Y}} = [k]$ (often $\widehat{\mathcal{Y}} = \mathcal{Y}$, but this need not always be the case). Given training examples $(X_1, Y_1), \ldots, (X_m, Y_m)$ drawn i.i.d. from some underlying distribution D on $\mathcal{X} \times [n]$, the goal is to learn a function $h : \mathcal{X} \to [k]$ with good performance according to some *loss function* $\ell : [n] \times [k] \to \mathbb{R}_+$, or equivalently, according to some *loss matrix* $\mathbf{L} \in \mathbb{R}_+^{n \times k}$ (we will use these two notions interchangeably, with the understanding that $L_{yt} = \ell(y,t)\ \forall y \in [n], t \in [k]$). In particular, the goal is to learn a function h with small *ℓ-generalization error* w.r.t. D, defined as $\mathrm{er}^{\ell}_D[h] = \mathbf{E}_{(X,Y)\sim D}[\ell(Y, h(X))]$; an algorithm that given m random examples learns a (random) function h_m is *ℓ-consistent* w.r.t. D if $\mathrm{er}^{\ell}_D[h_m] \xrightarrow{P} \inf_{h:\mathcal{X}\to[k]} \mathrm{er}^{\ell}_D[h]$ (as $m \to \infty$). For any $x \in \mathcal{X}$, we will denote $p_y(x) = \mathbf{P}(Y = y | X = x)\ \forall y \in [n]$ (under D) and $\mathbf{p}(x) = (p_1(x), \ldots, p_n(x))^\top$. For $\mathbf{p} \in \Delta_n$, we will find it convenient to define $\mathrm{Opt}(\ell, \mathbf{p}) = \mathrm{argmin}_{t \in [k]} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t)]$. Clearly, any classifier h that satisfies $h(x) \in \mathrm{Opt}(\ell, \mathbf{p}(x))\ \forall x \in \mathcal{X}$ achieves the optimal ℓ-error under D.

Surrogate risk minimization algorithms. Since minimizing the discrete loss ℓ directly is hard, a common algorithmic approach is to minimize a *surrogate loss* $\psi : [n] \times \mathbb{R}^d \to \mathbb{R}_+$ for some suitable $d \in \mathbb{Z}_+$. In particular, one learns a function $\mathbf{f}_m : \mathcal{X} \to \mathbb{R}^d$ by solving

$$\min_{\mathbf{f}} \sum_{i=1}^m \psi(Y_i, \mathbf{f}(X_i))$$

over a suitably rich class of functions $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$; and then returns $h_m = \text{pred} \circ \mathbf{f}_m$ for some suitable mapping $\text{pred} : \mathbb{R}^d \to [k]$ (for example, for multiclass 0-1 classification, where $k = n$ and $\ell_{\text{0-1}}(y,t) = \mathbf{1}(t \neq y)$, many common algorithms such as those considered by Zhang (2004b) and Tewari and Bartlett (2007) learn a function $\mathbf{f}_m : \mathcal{X} \to \mathbb{R}^n$ and then return a classifier $h_m = \text{argmax} \circ \mathbf{f}_m$). In practice, the surrogate ψ is often chosen to be convex in its second argument to enable efficient minimization. It is known that if the minimization is performed over a universal function class (with suitable regularization), then the resulting algorithm is ψ-consistent w.r.t. D, i.e. that the *ψ-generalization error* of $\mathbf{f}_m$ w.r.t. D, defined for a function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$ as $\text{er}_D^\psi[\mathbf{f}] = \mathbf{E}_{(X,Y)\sim D}[\psi(Y, \mathbf{f}(X))]$, converges to the optimal: $\text{er}_D^\psi[\mathbf{f}_m] \xrightarrow{P} \inf_{\mathbf{f}:\mathcal{X}\to\mathbb{R}^d} \text{er}_D^\psi[\mathbf{f}]$. There has been much work over the last several years on understanding when ψ-consistency (of $\mathbf{f}_m$) also implies ℓ-consistency (of h_m), and how to design surrogates satisfying this property; in particular, this has led to the study of surrogates that are *calibrated* with respect to the target loss ℓ (Bartlett et al., 2006; Zhang, 2004a,b; Tewari and Bartlett, 2007; Steinwart, 2007; Ramaswamy and Agarwal, 2012).

Calibrated surrogates. A pair (ψ, pred) is said to be *ℓ-calibrated* over $\mathcal{P} \subseteq \Delta_n$ if

$$\forall \mathbf{p} \in \mathcal{P} : \quad \inf_{\mathbf{u} \in \mathbb{R}^d : \text{pred}(\mathbf{u}) \notin \text{Opt}(\ell, p)} \mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y, \mathbf{u})] \;>\; \inf_{\mathbf{u}\in\mathbb{R}^d} \mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y, \mathbf{u})]\,. \tag{2.2.1}$$

It is known that (ψ, pred) is ℓ-calibrated over $\mathcal{P}$ if and only if ψ-consistency (of $\mathbf{f}_m$) implies ℓ-consistency (of $h_m = \text{pred} \circ \mathbf{f}_m$) for all distributions D for which $\mathbf{p}(x) \in \mathcal{P}\ \forall x$ (Bartlett et al., 2006; Zhang, 2004b; Tewari and Bartlett, 2007; Ramaswamy and Agarwal, 2012, 2015). Thus, given a target loss ℓ, in order to design a surrogate risk minimization algorithm that is ℓ-consistent w.r.t. some class of distributions D, one needs to design (ψ, pred) that is ℓ-calibrated over the corresponding set of conditional distributions $\mathcal{P}$. As noted above, one is often interested in *convex* calibrated surrogates, for which ψ is convex in its second argument, to enable efficient minimization.

2.2.2 Property Elicitation and Proper Scoring Rules/Losses

When the goal is to elicit a full distribution $\mathbf{p} \in \Delta_n$, it is well known that one can use a (strictly) proper scoring rule/loss. A scoring rule/loss in this context is a function $\psi : [n] \times \Delta_n \to \mathbb{R}_+$ that assigns a 'penalty' $\psi(y, \mathbf{p}')$ to an estimate/report $\mathbf{p}' \in \Delta_n$ when an outcome $y \in [n]$ is observed, and is said to be *proper* over $\mathcal{P} \subseteq \Delta_n$ if

$$\forall \mathbf{p} \in \mathcal{P} : \quad \mathbf{p} \ \in \ \operatorname{argmin}_{\mathbf{p}' \in \Delta_n} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{p}')] \,,$$

and *strictly proper* over $\mathcal{P}$ if the above minimizer is unique for all $\mathbf{p} \in \mathcal{P}$.[1] In probability forecasting and economics, where the goal is to elicit the distribution from an agent, the agent reports a distribution $\mathbf{p}'$, and on observing an outcome y drawn from the true distribution $\mathbf{p}$, receives a reward (or in our setting, incurs a loss) given by the scoring rule, namely $\psi(y, \mathbf{p}')$; a strictly proper scoring rule ensures that truthful reporting maximizes the agent's expected reward. In machine learning and statistics, where the goal is to estimate the distribution from random observations $y_1, \ldots, y_m$ sampled from $\mathbf{p}$, one estimates $\mathbf{p}'$ to minimize the average value of the scoring rule on the observed sample, $\frac{1}{m} \sum_{i=1}^{m} \psi(y_i, \mathbf{p}')$; here a strictly proper scoring rule yields a consistent estimator.

Proper (and strictly proper) scoring rules/losses for eliciting full probability distributions are fairly well characterized (Savage, 1971; Schervish, 1989; Gneiting and Raftery, 2007; Vernet et al., 2011). More recently, there has been much interest in understanding what types of scoring rules/losses can be used when the goal is to elicit not the full probability distribution $\mathbf{p}$, but rather some *property* of $\mathbf{p}$ of interest (Lambert et al., 2008; Lambert and Shoham, 2009; Abernethy and Frongillo, 2012; Steinwart et al., 2014; Frongillo and Kash, 2015).

Property of a distribution. In general, a property is any 'statistic' of a distribution.

[1] Note that we use the terms scoring rule and loss here interchangeably; in the literature, scoring rules usually assign a 'utility' to an estimate $\mathbf{p}'$ that needs to be maximized, while losses assign a 'penalty' that needs to be minimized. We will use the latter interpretation for both (in general, one can be obtained from the other simply by switching signs).

Formally, for $\mathcal{P} \subseteq \Delta_n$ and $d \in \mathbb{Z}_+$, we will define a (d-dimensional) *property* over $\mathcal{P}$ as any function $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ that maps each distribution $\mathbf{p} \in \mathcal{P}$ to a (d-dimensional) statistic $\Gamma(\mathbf{p}) \in \mathbb{R}^d$. One such example is the mean: $\Gamma(\mathbf{p}) = \mu(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[Y]$. Other examples of one-dimensional properties include the median, generalized quantiles, and many others. An example of a d-dimensional property is the vector of the first d moments: $\Gamma(\mathbf{p}) = (\mu_1(\mathbf{p}), \ldots, \mu_d(\mathbf{p}))^\top$, where $\mu_i(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[Y^i]$ $\forall i \in [d]$; more generally, a d-dimensional property is any vector of d one-dimensional properties.

Proper scoring rules/losses for eliciting properties of a distribution. Clearly, a (strictly) proper scoring rule that elicits the full distribution can be used to elicit any property of the distribution. However, this involves estimating an $(n-1)$-dimensional property, which can be expensive for large n and may not always be necessary. We will define a *d-dimensional scoring rule/loss* as a function $\psi : [n] \times \mathbb{R}^d \rightarrow \mathbb{R}_+$, and will say it is *proper for a property* $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ if

$$\forall \mathbf{p} \in \mathcal{P} : \qquad \Gamma(\mathbf{p}) \;\in\; \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^d} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})]\,,$$

and *strictly proper for* Γ if the above minimizer is unique for all $\mathbf{p} \in \mathcal{P}$. We will say a d-dimensional property $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ is *directly elicitable* if there exists a strictly proper d-dimensional scoring rule for Γ. Further, if for some $d' \geq d$, there is a directly elicitable d'-dimensional property $\Gamma' : \mathcal{P} \rightarrow \mathbb{R}^{d'}$ which can be used to recover Γ, i.e. for which there exists a mapping $\boldsymbol{\pi} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$ such that $\boldsymbol{\pi}(\Gamma'(\mathbf{p})) = \Gamma(\mathbf{p})$ $\forall \mathbf{p} \in \mathcal{P}$, then we will say that Γ is *d'-elicitable*. Clearly, every property is $(n-1)$-elicitable, and a d-dimensional property that is directly elicitable is d-elicitable.

Linear properties. A class of properties that are relatively better understood are *linear* properties. Specifically, a property $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ is said to be *linear* if it can be written as a vector of expectations, i.e. if there exists a function $\boldsymbol{\rho} : [n] \rightarrow \mathbb{R}^d$ such that $\Gamma(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[\boldsymbol{\rho}(Y)]$ $\forall \mathbf{p} \in \mathcal{P}$. It is known that linear properties are directly elicitable; moreover, as shown by Abernethy and Frongillo (2012), all strictly proper scoring rules for a linear property have the form of a Bregman divergence:

Theorem 2.2.1 (Abernethy and Frongillo (2012)). *Let $\mathcal{P} \subseteq \Delta_n$ and $\boldsymbol{\rho} : [n] \rightarrow \mathbb{R}^d$, and let $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ be a linear property defined as $\Gamma(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[\boldsymbol{\rho}(Y)]$ $\forall \mathbf{p} \in \mathcal{P}$. Then a scoring rule $\psi : [n] \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ is strictly proper for Γ if and only if there is a strictly convex function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ such that*

$$\psi(y, \mathbf{u}) = B_\phi(\boldsymbol{\rho}(y), \mathbf{u}) \quad \forall y \in [n], \mathbf{u} \in \mathbb{R}^d\,.$$

2.3 Calibrated Properties

We now make a connection between the two main themes of this chapter by defining the notion of a *calibrated property* for a given loss ℓ. As we will see, any strictly proper scoring rule for an ℓ-calibrated property will yield an ℓ-calibrated surrogate loss, and any ℓ-calibrated surrogate will yield a proper scoring rule for an ℓ-calibrated property.

Specifically, recall that given a loss $\ell : [n] \times [k] \rightarrow \mathbb{R}_+$, the goal is to learn a classifier that approaches the optimal ℓ-error under D, and that this is achieved by classifying according to $h(x) \in \mathrm{Opt}(\ell, \mathbf{p}(x))$ for all x. This means that for any $\mathbf{p} \in \Delta_n$ (or more generally, $\mathbf{p} \in \mathcal{P}$ for some suitable $\mathcal{P} \subseteq \Delta_n$), one is simply interested in finding an ℓ-optimal prediction $t^*(\mathbf{p}) \in [k]$, i.e. any $t^*(\mathbf{p})$ that satisfies $t^*(\mathbf{p}) \in \mathrm{argmin}_{t \in [k]} \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, t)]$. While we could consider the property $t^*(\mathbf{p})$ directly, this is a discrete-valued property that is generally hard to estimate directly.[2] Instead, we will consider properties $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$ that map $\mathbf{p} \in \mathcal{P}$ to a real number or vector $\Gamma(\mathbf{p}) \in \mathbb{R}^d$ from which one can *recover* an ℓ-optimal prediction $t^*(\mathbf{p}) \in [k]$ using a suitable mapping $\mathrm{pred} : \mathbb{R}^d \rightarrow [k]$; we will refer to such properties as *ℓ-calibrated* properties:

Definition 2.3.1 (ℓ-calibrated property). *Let $\mathcal{P} \subseteq \Delta_n$, $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^d$, and* $\mathrm{pred} : \mathbb{R}^d \rightarrow [k]$. *We*

[2] Note that in the probability forecasting/mechanism design setting, where there is an agent who holds information about the probability distribution and the goal is to elicit this information from him by assigning a suitable reward/loss using a scoring rule, eliciting a discrete-valued property poses no problem. However in the learning/statistics setting that we consider here, where one gets random observations from the underlying distribution and the goal is to estimate the property of interest from these observations by minimizing/maximizing a scoring rule, a discrete-valued property leads to a discrete optimization problem that in general can be hard.

will say (Γ, pred) *is* ℓ-calibrated *over* $\mathcal{P}$ *if for all* $\mathbf{p} \in \mathcal{P}$ *and all sequences* $\{\mathbf{u}_m\}$ *in* $\mathbb{R}^d$*,*

$$\mathbf{u}_m \to \Gamma(\mathbf{p}) \implies \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, \text{pred}(\mathbf{u}_m)] \to \min_{t\in[k]} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, t)]\,.$$

Note in particular this implies that if (Γ, pred) is ℓ-calibrated over $\mathcal{P}$, then we have that for all $\mathbf{p} \in \mathcal{P}$, $\text{pred}(\Gamma(\mathbf{p})) \in \text{Opt}(\ell, \mathbf{p})$. The sequence convergence condition is stronger and is needed in the proof of the following result, which tells us that the problem of designing an ℓ-calibrated surrogate loss in d dimensions can be reduced to finding an ℓ-calibrated property in d dimensions that is (directly) elicitable, together with any strictly proper scoring rule for it:

Theorem 2.3.1 (ℓ-calibrated surrogates via elicitable ℓ-calibrated properties)**.** *Let* $\ell : [n] \times [k]\to\mathbb{R}_+$ *and* $\mathcal{P} \subseteq \Delta_n$*. Let* $\Gamma : \mathcal{P}\to\mathbb{R}^d$ *and* $\text{pred} : \mathbb{R}^d\to[k]$ *be such that* Γ *is directly elicitable and* (Γ, pred) *is* ℓ*-calibrated over* $\mathcal{P}$*. Let* $\psi : [n] \times \mathbb{R}^d\to\mathbb{R}_+$ *be any strictly proper scoring rule for* Γ*. Then* (ψ, pred) *forms an* ℓ*-calibrated surrogate over* $\mathcal{P}$*.*

Proof. Let $\mathbf{p} \in \mathcal{P}$. By strict properness of ψ for Γ, we have that $\Gamma(\mathbf{p})$ is the unique minimizer of $\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y, \mathbf{u})]$ over $\mathbf{u} \in \mathbb{R}^d$; for convenience, denote this unique minimizer by $\mathbf{u}^*$. Now, for each $t \in [k]$, define

$$\text{regret}^{\ell}_{\mathbf{p}}(t) := \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, t)] - \min_{t\in[k]} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, t)]\,.$$

Since (Γ, pred) is ℓ-calibrated over $\mathcal{P}$, we have $\text{pred}(\mathbf{u}^*) = \text{pred}(\Gamma(\mathbf{p})) \in \text{Opt}(\ell, \mathbf{p})$, and therefore $\text{regret}^{\ell}_{\mathbf{p}}(\text{pred}(\mathbf{u}^*)) = 0$. Let

$$\epsilon = \min_{t\in[k]:\text{regret}^{\ell}_{\mathbf{p}}(t)>0} \text{regret}^{\ell}_{\mathbf{p}}(t)\,.$$

Then we have

$$\begin{aligned}\inf_{\mathbf{u}\in\mathbb{R}^d:\mathrm{pred}(\mathbf{u})\notin\mathrm{Opt}(\ell,\mathbf{p})}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})] &= \inf_{\mathbf{u}\in\mathbb{R}^d:\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}))\geq\epsilon}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})] \\ &= \inf_{\mathbf{u}\in\mathbb{R}^d:\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}))\geq\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}^*))+\epsilon}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]\,.\end{aligned}$$

Now, we claim that the mapping $\mathbf{u}\mapsto\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}))$ is continuous at $\mathbf{u}=\mathbf{u}^*$. To see this, note that since (Γ,pred) is ℓ-calibrated over $\mathcal{P}$, for all sequences $\{\mathbf{u}_m\}$ in $\mathbb{R}^d$ such that $\mathbf{u}_m\to\mathbf{u}^*$, we have $\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}_m))\to 0=\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}^*))$. In particular, this implies that $\exists\delta>0$ such that

$$\|\mathbf{u}-\mathbf{u}^*\|_2<\delta\implies\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}))-\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}^*))<\epsilon\,.$$

This

$$\begin{aligned}\inf_{\mathbf{u}\in\mathbb{R}^d:\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}))\geq\mathrm{regret}^{\ell}_{\mathbf{p}}(\mathrm{pred}(\mathbf{u}^*))+\epsilon}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})] &\geq \inf_{\mathbf{u}\in\mathbb{R}^d:\|\mathbf{u}-\mathbf{u}^*\|_2\geq\delta}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})] \\ &> \inf_{\mathbf{u}\in\mathbb{R}^d}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]\,.\end{aligned}$$

where the last inequality follows from the fact that $\mathbf{u}^*$ is the unique minimizer of $\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]$. Since $\mathbf{p}\in\mathcal{P}$ was arbitrary, the result follows. □

Theorem 2.3.2 (proper scoring rules via ℓ-calibrated surrogates)**.** *Let $\ell:[n]\times[k]\to\mathbb{R}_+$ and $\mathcal{P}\subseteq\Delta_n$. Let (ψ,pred) be an ℓ-calibrated surrogate where $\psi:[n]\times\mathbb{R}^d\to\mathbb{R}_+$ is continuous in the second argument and $\mathrm{pred}:\mathbb{R}^d\to[k]$. Then there exists an ℓ-calibrated property $\Gamma:\mathcal{P}\to\mathbb{R}^d$ over $\mathcal{P}$ such that ψ is a proper scoring rule for Γ over $\mathcal{P}$.*

Proof. Given $\mathbf{p}\in\mathcal{P}$, let $\mathbf{u}^*_{\mathbf{p}}\in\inf_{\mathbf{u}\in\mathbb{R}^d}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]$. We will consider the property $\Gamma:\mathcal{P}\to\mathbb{R}^d$ defined as $\Gamma(\mathbf{p}):=\mathbf{u}^*_{\mathbf{p}}$. It is easy to observe that ψ is a proper scoring rule for Γ since $\Gamma(\mathbf{p})=\mathbf{u}^*_{\mathbf{p}}\in\inf_{\mathbf{u}\in\mathbb{R}^d}\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]$ by definition. Hence, the rest of this proof is devoted to showing that Γ is ℓ-calibrated over $\mathcal{P}$.

We will first show that $\mathbf{u}^*_{\mathbf{p}}$ is such that $\mathrm{pred}(\mathbf{u}^*_{\mathbf{p}}) \in \mathrm{Opt}(\ell, \mathbf{p})$, for any $\mathbf{p} \in \mathcal{P}$. To see this suppose that $\mathrm{pred}(\mathbf{u}^*_{\mathbf{p}}) \notin \mathrm{Opt}(\ell, \mathbf{p})$, then we will have that

$$\inf_{\mathbf{u} \in \mathbb{R}^d : \mathrm{pred}(\mathbf{u}) \notin \mathrm{Opt}(\ell, p)} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] = \inf_{\mathbf{u} \in \mathbb{R}^d} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] \,,$$

which contradicts the definition of ℓ-calibration of surrogates (Eq. (2.2.1)).

Now, consider any sequence $\{\mathbf{u}_m\} \in \mathbb{R}^d$ such that $\mathbf{u}_m \to \mathbf{u}^*_{\mathbf{p}}$. We want to show that $\mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}_m))] \to \min_{t \in [k]} \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, t)]$. Equivalently, given any $\epsilon > 0$ we want to find $\delta > 0$ such that

$$\|\mathbf{u} - \mathbf{u}^*_{\mathbf{p}}\|_2 < \delta \implies \left| \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}))] \; - \; \min_{t \in [k]} \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, t)] \right| < \epsilon \,.$$

Let

$$\epsilon' := \inf_{\mathbf{u} \in \mathbb{R}^d : \mathrm{pred}(\mathbf{u}) \notin \mathrm{Opt}(\ell, \mathbf{p})} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] \; - \; \inf_{\mathbf{u} \in \mathbb{R}^d} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] \,. \tag{2.3.1}$$

Clearly, $\epsilon' > 0$ due to ℓ-calibration of ψ. The above implies that for any $\mathbf{u}$ with $\mathrm{pred}(\mathbf{u}) \notin \mathrm{Opt}(\ell, \mathbf{p})$ we have $\mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] - \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u}^*_{\mathbf{p}})] > \epsilon'$. Conversely,

$$\left| \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] - \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u}^*_{\mathbf{p}})] \right| < \epsilon' \implies \mathrm{pred}(\mathbf{u}) \in \mathrm{Opt}(\ell, \mathbf{p}) \,. \tag{2.3.2}$$

Since ψ is continuous at $\mathbf{u}^*_{\mathbf{p}} \in \mathbb{R}^d$, we know that for $\epsilon' > 0$ there exists a $\delta > 0$ such that

$$\|\mathbf{u} - \mathbf{u}^*_{\mathbf{p}}\|_2 < \delta \implies |\mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] \; - \; \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u}^*)]| < \epsilon' \,. \tag{2.3.3}$$

Using Eq. (2.3.2) and Eq. (2.3.3) one can observe that any $\mathbf{u}$ with $\|\mathbf{u} - \mathbf{u}^*_{\mathbf{p}}\|_2 < \delta$ is such that $\mathrm{pred}(\mathbf{u}) \in \mathrm{Opt}(\ell, \mathbf{p})$. Therefore, we have that

$$\|\mathbf{u} - \mathbf{u}^*_{\mathbf{p}}\|_2 < \delta \implies \left| \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}))] \; - \; \min_{t \in [k]} \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, t)] \right| = 0 < \epsilon \,.$$

This concludes the proof of sequence convergence requirement for ℓ-calibration in Definition 2.3.1. □

As a simple example, it is easy to see that $(n-1)$-dimensional properties that preserve the full probability structure (also called 'link' functions) are ℓ-calibrated for any loss ℓ, and that the corresponding strictly proper rules lead to class probability estimation (CPE) algorithms that estimate the full conditional distribution $\mathbf{p}(x)$ (and are consistent for any loss ℓ):

Example 2.3.2 (Link functions and class probability estimation (CPE))**.** *Let $\boldsymbol{\lambda} : \Delta_n \to \mathbb{R}^{n-1}$ be a bijective mapping (sometimes called a multiclass 'link' function) with a continuous inverse $\boldsymbol{\lambda}^{-1}$. Then the property $\Gamma : \Delta_n \to \mathbb{R}^{n-1}$ defined as $\Gamma(\mathbf{p}) = \boldsymbol{\lambda}(\mathbf{p})$ is trivially ℓ-calibrated over Δ_n for any loss $\ell : [n] \times [k] \to \mathbb{R}_+$; to see this, take any mapping $\mathrm{pred}_\ell : \mathbb{R}^{n-1} \to [k]$ that satisfies $\mathrm{pred}_\ell(\mathbf{u}) \in \mathrm{Opt}(\ell, \boldsymbol{\lambda}^{-1}(\mathbf{u}))\ \forall \mathbf{u} \in \mathbb{R}^{n-1}$. This property is also trivially elicitable; indeed, this is the property effectively elicited by class probability estimation algorithms using a multiclass proper composite surrogate loss with link $\boldsymbol{\lambda}$ (Vernet et al., 2011).*

While estimating the full conditional distribution $\mathbf{p}(x)$ clearly yields consistent algorithms for any loss ℓ, this requires $n-1$ dimensions and is not always needed. Indeed, for many losses ℓ, finding an optimal classifier requires estimating only a restricted, lower-dimensional property of $\mathbf{p}(x)$. In such cases, one can use a strictly proper scoring rule for the corresponding property to design a calibrated surrogate loss operating in a smaller number of dimensions. We shall see several examples of such surrogates below. In particular, in Section 2.4 we shall see examples of calibrated surrogate losses that effectively elicit low-dimensional linear properties of $\mathbf{p}(x)$. In Section 2.5 we will consider how to exploit low-dimensional nonlinear calibrated properties. In both cases, we will be particularly interested in *convex* scoring rules that lead to convex calibrated surrogates.

2.4 Calibrated Surrogates via Calibrated Linear Properties

In this section we show that some recent works that have proposed general frameworks for obtaining convex calibrated surrogates effectively amount to using proper scoring rules for calibrated linear properties. In particular, we start by showing that the notion of 'standardization function' used to obtain calibrated surrogates for certain subset ranking losses (Buffoni et al., 2011) corresponds to a calibrated linear property (Section 2.4.1). We then show that the general framework described recently by Ramaswamy et al. (2013) for obtaining convex calibrated surrogates for any loss ℓ in $d = \text{affdim}(\mathbf{L})$ dimensions also amounts to using a calibrated linear property (Section 2.4.2). Finally, we show that for any loss ℓ, the number of dimensions d needed to construct an ℓ-calibrated linear property is fundamentally lower bounded by $\text{affdim}(\mathbf{L}) - 1$ (Section 2.4.3), making the construction of Ramaswamy et al. (2013) essentially unimprovable as far as linear properties are concerned.

2.4.1 Subset Ranking Losses and Standardization Functions

Subset ranking refers to ranking problems such as those that arise in information retrieval, where each instance $x \in \mathcal{X}$ consists of a query with say r associated documents, and a label $y \in \mathcal{Y}$ represents some 'preference' or 'relevance' information about these documents in relation to the query; for example a label could be a (possibly weighted) directed acyclic graph (DAG) on r nodes indicating which of the r documents are more relevant to the query than others ($\mathcal{Y} = \mathcal{G}_r$ for some finite set $\mathcal{G}_r$ of possibly weighted DAGs on r nodes, with $n = |\mathcal{G}_r|$), or simply a vector of r binary or multi-valued relevance judgments for the documents ($\mathcal{Y} = \{0,1\}^r$ with $n = 2^r$ or $\mathcal{Y} = [q]^r$ for some $q \in \mathbb{Z}_+$ with $n = q^r$). In most such settings, given a new query with r documents, the goal is to rank the documents by relevance to the query, i.e. the prediction space is the set of permutations of r objects, $\widehat{\mathcal{Y}} = \mathcal{S}_r$ (thus $k = r!$). There has been much work in recent years on understanding how to design convex calibrated surrogates for various subset ranking losses used in practice, such as the (normalized) discounted cumulative gain ((N)DCG), pairwise disagreement (PD), mean average precision (MAP), etc (Cossock and Zhang, 2008; Xia et al., 2008; Duchi et al., 2010; Ravikumar et al., 2011; Buffoni et al., 2011; Calauzènes et al., 2012; Lan et al., 2012).

In particular, Buffoni et al. (2011) introduced the notion of 'standardization function', and showed that many previous results on calibrated surrogates for subset ranking could be explained through this notion. Specifically, let $\mathcal{Y}$ be one of the label spaces above and $\widehat{\mathcal{Y}} = \mathcal{S}_r$, and let $\ell : \mathcal{Y} \times \widehat{\mathcal{Y}} \rightarrow \mathbb{R}_+$ be any subset ranking loss. A *standardization function* for ℓ over $\mathcal{P} \subseteq \Delta_{\mathcal{Y}}$ is defined as any function $\mathbf{s} : \mathcal{Y} \rightarrow \mathbb{R}^r$ such that

$$\forall \mathbf{p} \in \mathcal{P} : \qquad \mathrm{argsort}\big(\mathbf{E}_{Y\sim\mathbf{p}}[\mathbf{s}(Y)]\big) \ \subseteq\ \mathrm{argmin}_{\sigma\in\mathcal{S}_r} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,\sigma)]\,. \tag{2.4.1}$$

We show below that if such a function $\mathbf{s}$ exists, then the r-dimensional linear property $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^r$ defined as $\Gamma(\mathbf{p}) = \mathbf{E}_{Y\sim\mathbf{p}}[\mathbf{s}(Y)]$ is ℓ-calibrated over $\mathcal{P}$:

Theorem 2.4.1 (Standardization functions yield calibrated linear properties)**.** *Let $\ell : \mathcal{Y} \times \mathcal{S}_r \rightarrow \mathbb{R}_+$ be a subset ranking loss for some suitable $\mathcal{Y}$ as above, and let $\mathcal{P} \subseteq \Delta_{\mathcal{Y}}$. Let $\mathbf{s} : \mathcal{Y} \rightarrow \mathbb{R}^r$ be a standardization function for ℓ over $\mathcal{P}$. Let $\Gamma : \mathcal{P} \rightarrow \mathbb{R}^r$ be the linear property defined as*

$$\Gamma(\mathbf{p}) = \mathbf{E}_{Y\sim\mathbf{p}}[\mathbf{s}(Y)]\,,$$

and let $\mathrm{pred} : \mathbb{R}^r \rightarrow \mathcal{S}_r$ *be any mapping that satisfies* $\mathrm{pred}(\mathbf{u}) \in \mathrm{argsort}(\mathbf{u})\ \forall \mathbf{u} \in \mathbb{R}^r$*. Then* (Γ, pred) *is ℓ-calibrated over $\mathcal{P}$.*

Proof. Let $\mathbf{p} \in \mathcal{P}$, and let $\{\mathbf{u}_m\}$ be any sequence in $\mathbb{R}^r$ such that $\mathbf{u}_m \rightarrow \Gamma(\mathbf{p})$. We will show that $\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}_m))] \rightarrow \min_{\sigma\in S_r} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,\sigma)]$.

Let $\delta := \min_{i,j\in[r]:|\Gamma_i(\mathbf{p})-\Gamma_j(\mathbf{p})|>0} |\Gamma_i(\mathbf{p}) - \Gamma_j(\mathbf{p})|$. Since $\mathbf{u}_m \rightarrow \Gamma(p)$, we have $\exists M$ such that

$$\forall m \geq M : \quad \|\mathbf{u}_m - \Gamma(\mathbf{p})\|_2 < \delta\,.$$

Now clearly, for all $m \geq M$ and $i, j \in [r]$, we must have $\Gamma_i(\mathbf{p}) > \Gamma_j(\mathbf{p}) \implies u_{mi} > u_{mj}$ (else the L_2-distance between $\mathbf{u}_m$ and $\Gamma(\mathbf{p})$ would exceed δ). Therefore, for all $m \geq M$, we have $\mathrm{argsort}(\mathbf{u}_m) \subseteq \mathrm{argsort}(\Gamma(\mathbf{p}))$, and thus $\mathbf{E}_{Y\sim\mathbf{p}}\big[\ell(Y, \mathrm{pred}(\mathbf{u}_m))\big] = \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, \mathrm{pred}(\Gamma(\mathbf{p})))]$. Also, by construction of pred, we know that $\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, \mathrm{pred}(\Gamma(\mathbf{p})))] = \min_{\sigma\in S_r} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,\sigma)]$. This implies that for all $m \geq M$, $\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}_m))] = \min_{\sigma\in S_r} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,\sigma)]$.

Since $\mathbf{p} \in \mathcal{P}$ was arbitrary, this proves the result. □

Thus, if a subset ranking loss ℓ has a standardization function over $\mathcal{P}$, then one can construct an r-dimensional convex calibrated surrogate for ℓ over $\mathcal{P}$ by constructing a convex strictly proper scoring rule for the calibrated linear property Γ above (e.g. by using $\phi(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_2^2$ in Theorem 2.2.1). Note that this is a huge savings over the naïve CPE approach of Example 2.3.2, which would use $|\mathcal{Y}| - 1$ dimensions (for most subset ranking settings, $|\mathcal{Y}|$ is exponential in r). The following example illustrates one application of the above result:

Example 2.4.1 (Discounted cumulative gain (DCG) loss for subset ranking)**.** *The DCG loss for multi-valued relevance vector labels ($\mathcal{Y} = [q]^r$ for some $q \in \mathbb{Z}_+$), $\ell_{\mathrm{DCG@}\tau} : [q]^r \times \mathcal{S}_r \to \mathbb{R}_+$ (where $\tau \in [r]$ is a cut-off value), is widely used in information retrieval and is defined as*

$$\ell_{\mathrm{DCG@}\tau}(\mathbf{y}, \sigma) \;=\; Z - \sum_{i=1}^{\tau} \frac{2^{y_{\sigma^{-1}(i)}} - 1}{\log_2(i+1)} \qquad \forall \mathbf{y} \in [q]^r, \sigma \in \mathcal{S}_r$$

for a suitable constant Z that ensures non-negativity of the loss. As shown by Buffoni et al. (2011), the function $\mathbf{s} : [q]^r \to \mathbb{R}^r$ defined as $s_i(\mathbf{y}) = 2^{y_{\sigma^{-1}(i)}} - 1 \ \forall i \in [r]$ is a standardization function for $\ell_{\mathrm{DCG@}\tau}$ over $\Delta_{\mathcal{Y}}$, and therefore it follows from Theorem 2.4.1 that any strictly proper scoring rule for the corresponding linear property $\Gamma : \Delta_{\mathcal{Y}} \to \mathbb{R}^r$ given by $\Gamma_i(\mathbf{p}) = \mathbf{E}_{\mathbf{Y} \sim \mathbf{p}}[2^{Y_{\sigma^{-1}(i)}} - 1] \ \forall i \in [r], \mathbf{p} \in \Delta_{\mathcal{Y}}$ yields an $\ell_{\mathrm{DCG@}\tau}$-calibrated surrogate over $\Delta_{\mathcal{Y}}$. In particular, using $\phi(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_2^2$ in Theorem 2.2.1, one gets the convex $\ell_{\mathrm{DCG@}\tau}$-calibrated surrogate used by Cossock and Zhang (2008).

Another example of an application of Theorem 2.4.1 involves the weighted pairwise disagreement (WPD) loss for subset ranking (Duchi et al., 2010). In particular, Duchi et al. (2010) proposed a convex r-dimensional surrogate for subset ranking which they showed to be calibrated w.r.t. the WPD loss under a certain low-noise condition; this surrogate can also be viewed as a strictly proper scoring rule for a linear property, composed with a link function.

Example 2.4.2 (Weighted pairwise disagreement (WPD) loss for subset ranking)**.** *Another popular subset ranking loss is the WPD loss for weighted preference graph labels, $\ell_{\text{WPD}} : \mathcal{Y} \times \mathcal{S}_r \to \mathbb{R}_+$, where $\mathcal{Y}$ is some finite set of weighted DAGs on r nodes; for a weighted DAG $G = ([r], E^G, \mathbf{W}^G) \in \mathcal{Y}$, where $E^G \subset [r] \times [r]$ denotes the set of edges of G and $\mathbf{W}^G \in \mathbb{R}_+^{r \times r}$ denotes the edge weights with $W_{ij}^G > 0$ iff $(i,j) \in E^G$, and for a permutation $\sigma \in \mathcal{S}_r$, this loss is defined as*

$$\ell_{\text{WPD}}(G, \sigma) \;=\; \sum_{i,j} W_{ij}^G \Big(\mathbf{1}(\sigma(i) > \sigma(j)) + \frac{1}{2} \mathbf{1}(\sigma(i) = \sigma(j)) \Big) \,.$$

For any $\mathbf{p} \in \Delta_{\mathcal{Y}}$, define $W_{ij}^{\mathbf{p}} = \mathbf{E}_{G \sim \mathbf{p}}[W_{ij}^G]$ and $E^{\mathbf{p}} = \big\{(i,j) \in [r] \times [r] : W_{ij}^{\mathbf{p}} > W_{ji}^{\mathbf{p}}\big\}$. Duchi et al. (2010) considered the following set of 'low-noise' distributions $\mathbf{p} \in \Delta_{\mathcal{Y}}$:

$$\begin{aligned} \mathcal{P}_{\text{LN}}^{\text{WPD}} \;=\; & \Big\{ \mathbf{p} \in \Delta_{\mathcal{Y}} : \textit{the unweighted graph } G^{\mathbf{p}} = ([r], E^{\mathbf{p}}) \textit{ is a DAG, and} \\ & \quad \forall i, k \in [r] : \; W_{ik}^{\mathbf{p}} > W_{ki}^{\mathbf{p}} \implies \textstyle\sum_{j=1}^r \big(W_{ij}^{\mathbf{p}} - W_{ji}^{\mathbf{p}}\big) > \sum_{j=1}^r \big(W_{kj}^{\mathbf{p}} - W_{jk}^{\mathbf{p}}\big) \Big\} \,. \end{aligned}$$

It is easy to see that the function $\mathbf{s} : \mathcal{Y} \to \mathbb{R}^r$ defined as $s_i(G) = \sum_{j=1}^r (W_{ij}^G - W_{ji}^G) \; \forall i \in [r]$ is a standardization function for ℓ_{WPD} over $\mathcal{P}_{\text{LN}}^{\text{WPD}}$, and therefore by Theorem 2.4.1, any strictly proper scoring rule for the corresponding linear property $\Gamma : \mathcal{P}_{\text{LN}}^{\text{WPD}} \to \mathbb{R}^r$ given by $\Gamma_i(\mathbf{p}) = \sum_{j=1}^r (W_{ij}^{\mathbf{p}} - W_{ji}^{\mathbf{p}}) \; \forall i \in [r], \mathbf{p} \in \mathcal{P}_{\text{LN}}^{\text{WPD}}$ yields an ℓ_{WPD}-calibrated surrogate over $\mathcal{P}_{\text{LN}}^{\text{WPD}}$. The convex r-dimensional surrogate shown to be ℓ_{WPD}-calibrated over $\mathcal{P}_{\text{LN}}^{\text{WPD}}$ by Duchi et al. (2010) can be viewed as a strictly proper scoring rule for this property composed with a link function.

2.4.2 Affdim(L)-Dimensional Surrogates of Ramaswamy et al. (2013)

Recently, Ramaswamy et al. (2013) gave a very general framework for constructing a convex calibrated surrogate (over the full simplex Δ_n) for any given loss $\ell : [n] \times [k] \to \mathbb{R}_+$ in $d = \text{affdim}(\mathbf{L})$ dimensions. In particular, they gave the following result:

Theorem 2.4.2 (Ramaswamy et al. (2013))**.** *Let $\ell : [n] \times [k] \to \mathbb{R}_+^k$ be such that $\mathbf{L} = \mathbf{AB} + c$ for some $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{B} \in \mathbb{R}^{d \times k}$, and $c \in \mathbb{R}$. Let $\psi : [n] \times \mathbb{R}^d \to \mathbb{R}_+$ and* pred $: \mathbb{R}^d \to [k]$ *be* *defined as follows:*

$$\psi(y, \mathbf{u}) = \textstyle\sum_{i=1}^{d} (u_i - A_{yi})^2 \,, \qquad \mathrm{pred}(\mathbf{u}) \in \mathrm{argmin}_{t \in [k]} \textstyle\sum_{i=1}^{d} B_{it} u_i \,.$$

Then (ψ, pred) *is* ℓ*-calibrated over* Δ_n*.*

The proof of the above result (Ramaswamy et al., 2013) can be re-interpreted as showing that the linear property $\Gamma : \Delta_n \rightarrow \mathbb{R}^d$ (where $d = \mathrm{affdim}(\mathbf{L})$) given by $\Gamma_i(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[A_{Yi}]\ \forall i \in [d]$ is ℓ-calibrated over Δ_n via the above mapping pred; the convex least-squares type surrogate loss ψ defined above is then simply the strictly proper scoring rule for this property resulting from using $\phi(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_2^2$ in Theorem 2.2.1. For completeness, we state this below and give a self-contained proof. Note also that this implies that any other strictly proper scoring rule for this linear property (such as those obtained by using Bregman divergences associated with other convex functions ϕ in Theorem 2.2.1) will also lead to an ℓ-calibrated surrogate over Δ_n.

Theorem 2.4.3 (Affdim($\mathbf{L}$)-dimensional calibrated linear properties)**.** *Let* $\ell : [n] \times [k] \rightarrow \mathbb{R}_+^k$ *be such that* $\mathbf{L} = \mathbf{AB} + c$ *for some* $\mathbf{A} \in \mathbb{R}^{n \times d}$*,* $\mathbf{B} \in \mathbb{R}^{d \times k}$*, and* $c \in \mathbb{R}$*. Let* $\Gamma : \Delta_n \rightarrow \mathbb{R}^d$ *be the linear property defined as*

$$\Gamma_i(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[A_{Yi}] \qquad \forall i \in [d],$$

and let $\mathrm{pred} : \mathbb{R}^d \rightarrow [k]$ *be defined as in Theorem 2.4.2. Then* (Γ, pred) *is* ℓ*-calibrated over* Δ_n*.*

Proof. Note first that for any $\mathbf{p} \in \Delta_n$ and $t \in [k]$, we have

$$\begin{aligned}
\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t)] &= \sum_{y=1}^{d} p_y \Big(\sum_{i=1}^{d} A_{yi} B_{it} + c \Big) \\
&= \sum_{y=1}^{d} \sum_{i=1}^{d} p_y A_{yi} B_{it} + c \\
&= \sum_{i=1}^{d} B_{it} \sum_{y=1}^{d} p_y A_{yi} + c \\
&= \sum_{i=1}^{d} B_{it} \mathbf{E}_{Y\sim\mathbf{p}}[A_{Yi}] + c \;=\; \sum_{i=1}^{d} B_{it} \Gamma_i(\mathbf{p}) + c\,. \qquad (2.4.2)
\end{aligned}$$

Now, let $\mathbf{p} \in \Delta_n$, and let $\{\mathbf{u}_m\}$ be any sequence in $\mathbb{R}^d$ such that $\mathbf{u}_m \to \Gamma(\mathbf{p})$. For each m, define $t_m := \mathrm{pred}(\mathbf{u}_m) \in [k]$. Then we have

$$\begin{aligned}
&\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t_m)] - \min_{t\in[k]} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t)] \\
&= \sum_{i=1}^{d} B_{it_m} \Gamma_i(\mathbf{p}) - \min_{t\in[k]} \sum_{i=1}^{d} B_{it} \Gamma_i(\mathbf{p})\,, \qquad \text{by Eq. (2.4.2)} \\
&= \sum_{i=1}^{d} B_{it_m} (\Gamma_i(\mathbf{p}) - u_{mi}) + \sum_{i=1}^{d} B_{it_m} u_{mi} - \min_{t\in[k]} \sum_{i=1}^{d} B_{it} \Gamma_i(\mathbf{p}) \\
&= \sum_{i=1}^{d} B_{it_m} (\Gamma_i(\mathbf{p}) - u_{mi}) + \min_{t\in[k]} \sum_{i=1}^{d} B_{it} u_{mi} - \min_{t\in[k]} \sum_{i=1}^{d} B_{it} \Gamma_i(\mathbf{p})\,,
\end{aligned}$$

where the last equality holds due to the definition of pred. It is easy to see that the term on the right hand side goes to zero as $m\to\infty$. Thus we get that $\mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t_m)] \to \min_{t\in[k]} \mathbf{E}_{Y\sim\mathbf{p}}[\ell(Y,t)]$. Since $\mathbf{p} \in \Delta_n$ was arbitrary, this proves the result. □

Ramaswamy et al. (2013) also applied Theorem 2.4.2 to obtain low-dimensional convex calibrated surrogates for several subset ranking losses. For subset ranking losses with $\mathrm{affdim}(\mathbf{L}) = r$ (such as the DCG@r loss), the linear property constructed by the above result effectively provides a standardization function over $\Delta_{\mathcal{Y}}$. For other subset ranking losses, the two approaches can give complementary results. For example, for the WPD and MAP losses, which have affine dimensions $\Theta(r^2)$ (Ramaswamy and Agarwal, 2015), it is known that there is no standardization function over $\Delta_{\mathcal{Y}}$ (Buffoni et al., 2011), and that there is no convex calibrated surrogate over $\Delta_{\mathcal{Y}}$ in r dimensions (Calauzènes et al., 2012; Ramaswamy and Agarwal, 2015). On the other hand, by Theorem 2.4.2, there do exist $\Theta(r^2)$-dimensional calibrated linear properties and therefore $\Theta(r^2)$-dimensional convex calibrated surrogates for these losses over $\Delta_{\mathcal{Y}}$; moreover, as demonstrated in Example 2.4.2, one can construct standardization functions for these losses over restricted sets of distributions $\mathcal{P} \subset \Delta_{\mathcal{Y}}$, allowing for r-dimensional convex calibrated surrogates over such restricted sets $\mathcal{P}$.

The following example illustrates a different application of the above result:

Example 2.4.3 (Hamming loss for sequence prediction)**.** *Consider a sequence prediction task with $\mathcal{Y} = \widehat{\mathcal{Y}} = \{0,1\}^r$ (thus $n = k = 2^r$). A widely used loss in this setting is the Hamming loss $\ell_{\text{Ham}} : \{0,1\}^r \times \{0,1\}^r \to \mathbb{R}_+$ given by*

$$\ell_{\text{Ham}}(\mathbf{y},\mathbf{t}) = \textstyle\sum_{i=1}^r \mathbf{1}(t_i \neq y_i) \qquad \forall \mathbf{y},\mathbf{t} \in \{0,1\}^r \,.$$

As shown by Ramaswamy and Agarwal (2012), $\text{affdim}(\mathbf{L}^{\text{Ham}}) \leq r$*, and therefore by Theorem 2.4.3, one can construct an r-dimensional linear property $\Gamma : \Delta_{\mathcal{Y}} \to \mathbb{R}^r$ that is ℓ_{Ham}-calibrated over $\Delta_{\mathcal{Y}}$. Any strictly proper scoring rule for Γ then forms an r-dimensional ℓ_{Ham}-calibrated surrogate over $\Delta_{\mathcal{Y}}$; in particular, using $\phi(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_2^2$ in Theorem 2.2.1, one gets the surrogate given by Theorem 2.4.2.*

2.4.3 Lower Bound on Dimension of Calibrated Linear Properties

Theorem 2.4.3 shows that for any loss ℓ, there is a linear property in $d = \text{affdim}(\mathbf{L})$ dimensions that is ℓ-calibrated over Δ_n. In the following result, we show that this is essentially the best one can do with linear properties:

Theorem 2.4.4 (Lower bound on dimension of calibrated linear properties)**.** *Let $\ell : [n] \times [k] \to \mathbb{R}_+$. Let $\Gamma : \Delta_n \to \mathbb{R}^d$ be a linear property. If there exists a mapping* $\text{pred} : \mathbb{R}^d \to [k]$ *such* *that* (Γ, pred) *is* ℓ*-calibrated over* Δ_n*, then*

$$d \;\geq\; \mathrm{affdim}(\mathbf{L}) - 1\,.$$

Proof. For each $t \in [k]$, denote $\boldsymbol{\ell}_t = (\ell(1,t), \cdots, \ell(n,t))^\top$. Before proceeding with the proof, we will need the following definition of trigger probabilities:

Definition 2.4.4 (Trigger Probabilities; Ramaswamy and Agarwal (2012))**.** *Let* $\ell : [n] \times [k] \to \mathbb{R}_+$*. For each* $t \in [k]$*, the set of trigger probabilities of* t *with respect to* ℓ *is defined as*

$$\mathcal{Q}^\ell_t := \big\{\mathbf{p} \in \Delta_n : \mathbf{p}^\top(\boldsymbol{\ell}_t - \boldsymbol{\ell}_{t'}) \leq 0 \;\; \forall t' \in [k]\big\} = \big\{\mathbf{p} \in \Delta_n : t \in \mathrm{Opt}(\ell, \mathbf{p})\big\}\,.$$

Suppose $\exists \mathrm{pred} : \mathbb{R}^d \to [k]$ such that (Γ, pred) is ℓ-calibrated over Δ_n. We will show that $d \geq \mathrm{affdim}(\mathbf{L}) - 1$.

Suppose for the sake of contradiction that $d < \mathrm{affdim}(\mathbf{L}) - 1$. Let $\mathbf{s} : [n] \to \mathbb{R}^d$ be such that $\Gamma(\mathbf{p}) = \mathbf{E}_{Y \sim \mathbf{p}}[\mathbf{s}(Y)] \; \forall \mathbf{p} \in \Delta_n$, and define $\mathbf{U} \in \mathbb{R}^{d \times n}$ as $u_{iy} := s_i(y) \; \forall i \in [d], y \in [n]$. Observe that $\Gamma(\mathbf{p}) = \mathbf{U}\mathbf{p}$. For each $i \in [d]$, let $\mathbf{u}_i \in \mathbb{R}^n$ denote the i-th row vector of $\mathbf{U}$, so that $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_d]^\top$. Define $\widetilde{\mathbf{U}} := [\mathbf{u}_1 \cdots \mathbf{u}_d \, \mathbf{1}]^\top$, where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector.

The main idea of the proof is to find $\mathbf{p}_1, \mathbf{p}_2 \in \Delta_n$ such that $\mathbf{U}\mathbf{p}_1 = \mathbf{U}\mathbf{p}_2$ but $\mathrm{Opt}(\ell, \mathbf{p}_1) \cap \mathrm{Opt}(\ell, \mathbf{p}_2) = \emptyset$; this will contradict the fact that (Γ, pred) is ℓ-calibrated over Δ_n. We find such $\mathbf{p}_1, \mathbf{p}_2$ by first finding $\mathbf{p} \in \Delta_n$ that lies at the intersection of two trigger probability sets, and then perturbing it along suitable directions $\boldsymbol{\delta}, -\boldsymbol{\delta}$ (see Figure 1). The following steps give more details.

Step 1: Let $i, j \in [k]$ be such that $\boldsymbol{\ell}_i - \boldsymbol{\ell}_j \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$ and $\mathcal{Q}^\ell_i \cap \mathcal{Q}^\ell_j \neq \emptyset$. To see that such i, j always exist, note that by our assumption that $d + 1 < \mathrm{affdim}(\mathbf{L})$, $\exists i', j' \in [k]$ such that $\boldsymbol{\ell}_{i'} - \boldsymbol{\ell}_{j'} \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$. If $\mathcal{Q}^\ell_{i'} \cap \mathcal{Q}^\ell_{j'} \neq \emptyset$, define $i := i'$ and $j := j'$ and we are done. Suppose that $\mathcal{Q}^\ell_{i'} \cap \mathcal{Q}^\ell_{j'} = \emptyset$. Consider a sequence of neighboring trigger probability sets

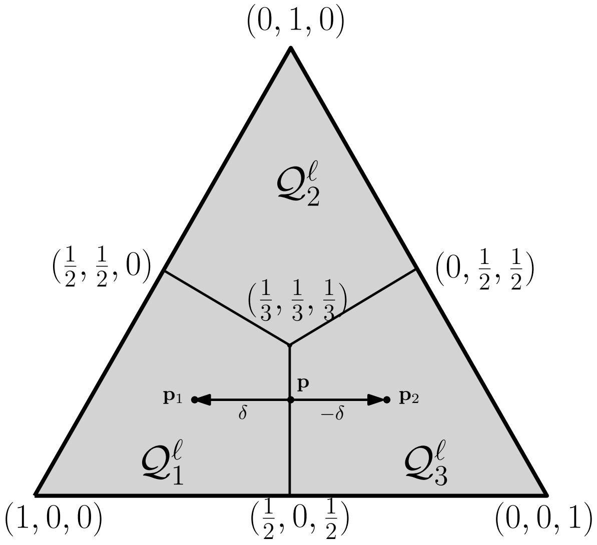


Figure 1: Illustration of steps in the proof of Theorem 2.4.4. We first find $\mathbf{p} \in \mathcal{Q}_1^\ell \cap \mathcal{Q}_3^\ell$, and then perturb $\mathbf{p}$ along $\boldsymbol{\delta}$ and $-\boldsymbol{\delta}$ to find $\mathbf{p}_1$ and $\mathbf{p}_2$.

$\mathcal{Q}_{i_1}^\ell, \mathcal{Q}_{i_2}^\ell, \cdots, \mathcal{Q}_{i_m}^\ell$ such that $i_1 = i'$, $i_m = j'$, and $\mathcal{Q}_{i_r}^\ell \cap \mathcal{Q}_{i_{r+1}}^\ell \neq \emptyset$ for all $r \in [m-1]$. We can write $\boldsymbol{\ell}_{i'} - \boldsymbol{\ell}_{j'} = (\boldsymbol{\ell}_{i_1} - \boldsymbol{\ell}_{i_2}) + (\boldsymbol{\ell}_{i_2} - \boldsymbol{\ell}_{i_3}) + \cdots + (\boldsymbol{\ell}_{i_{m-1}} - \boldsymbol{\ell}_{i_m})$. Since $\boldsymbol{\ell}_{i'} - \boldsymbol{\ell}_{j'} \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$, $\exists r \in [m-1]$ such that $\boldsymbol{\ell}_{i_r} - \boldsymbol{\ell}_{i_{r+1}} \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$. Define $i := r$ and $j := r+1$. Then we have $\boldsymbol{\ell}_i - \boldsymbol{\ell}_j \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$ and $\mathcal{Q}_i^\ell \cap \mathcal{Q}_j^\ell \neq \emptyset$.

Step 2: Fix i, j as above, and let $\mathbf{p} \in \mathcal{Q}_i^\ell \cap \mathcal{Q}_j^\ell \cap \mathrm{relint}(\Delta_n)$ such that $\mathbf{p} \notin \mathcal{Q}_t^\ell\ \forall t \neq i, j$ (which means that $\mathbf{p}^\top \boldsymbol{\ell}_i = \mathbf{p}^\top \boldsymbol{\ell}_j < \mathbf{p}^\top \boldsymbol{\ell}_t\ \forall t \neq i, j$). The trigger probability sets form a power diagram of the probability simplex, which implies that $\mathcal{Q}_i^\ell \cap \mathcal{Q}_j^\ell \not\subset \mathrm{bndry}(\Delta_n)$ and $\mathcal{Q}_i^\ell \cap \mathcal{Q}_j^\ell \not\subset \mathcal{Q}_t^\ell\ \forall t \neq i, j$; therefore, such a point $\mathbf{p}$ always exists.

Step 3: Let $\boldsymbol{\delta} \in \mathbb{R}^n$ such that $\widetilde{\mathbf{U}}\boldsymbol{\delta} = \mathbf{0}$ and $(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta} \neq 0$. To see that such a $\boldsymbol{\delta}$ always exists, let $p = \mathrm{rank}(\widetilde{\mathbf{U}})$. Observe that $p < n-1$ as $d < \mathrm{affdim}(\mathbf{L}) - 1$ and $p \leq d$. Let $\mathbf{v}_1, \cdots, \mathbf{v}_{n-p} \in \mathbb{R}^n$ be an orthonormal basis of the null space of $\widetilde{\mathbf{U}}$. Clearly, $\mathrm{span}(\mathbf{u}_1, \cdots, \mathbf{u}_d, \mathbf{1}, \mathbf{v}_1, \cdots, \mathbf{v}_{n-p}) = \mathbb{R}^n$, and therefore, $\exists \alpha_1, \cdots, \alpha_{d+1}, \beta_1, \cdots, \beta_{n-p}$ such that $\boldsymbol{\ell}_i - \boldsymbol{\ell}_j = \sum_{r=1}^d \alpha_r \mathbf{u}_r + \alpha_{d+1}\mathbf{1} + \sum_{r=1}^{n-p} \beta_r \mathbf{v}_r$. Since $\boldsymbol{\ell}_i - \boldsymbol{\ell}_j \notin \mathrm{col}(\widetilde{\mathbf{U}}^\top)$, $\exists q \in [n-p]$ such that

$\beta_q \neq 0$. Take $\boldsymbol{\delta} = \mathbf{v}_q$. By construction, $\widetilde{\mathbf{U}}\boldsymbol{\delta} = \mathbf{0}$. Moreover,

$$\begin{aligned}(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta} &= \sum_{r=1}^{d} \alpha_r \mathbf{u}_r^\top \mathbf{v}_q + \alpha_{d+1} \mathbf{1}^\top \mathbf{v}_q + \sum_{r=1}^{n-p} \beta_r \mathbf{v}_r^\top \mathbf{v}_q \\ &= \beta_q ||\mathbf{v}_q||_2^2 \,, \qquad\qquad \text{since } \widetilde{\mathbf{U}}\mathbf{v}_q = 0 \text{ and } \mathbf{v}_r^\top \mathbf{v}_q = 0 \; \forall r \neq q \\ &\neq 0 \,.\end{aligned}$$

Thus we have shown that $\exists \boldsymbol{\delta} \in \mathbb{R}^n$ such that $\widetilde{\mathbf{U}}\boldsymbol{\delta} = \mathbf{0}$ and $(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta} \neq 0$. In the remainder of the proof we will assume without loss of generality that $(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta} < 0$ (the case $(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta} > 0$ can be treated similarly as below).

Step 4: This is the most crucial step in the proof in which we find $\mathbf{p}_1, \mathbf{p}_2$ by perturbing $\mathbf{p}$ along $\boldsymbol{\delta}$ as shown in Figure 1. We have to ensure: (1) This perturbation leads to valid probability vectors; (2) One of the perturbed vectors lands in $\mathcal{Q}_i^\ell$ and the other one lands in $\mathcal{Q}_j^\ell$.

Let a be the least positive integer such that $\forall r \in [n]$, $|\delta_r/a| \leq \min(p_r, 1-p_r)$, and let $\boldsymbol{\delta}' := \boldsymbol{\delta}/a$. Next, let b be the least positive integer such that $\forall t \neq i, j$,

$$\mathbf{p}^\top(\boldsymbol{\ell}_t - \boldsymbol{\ell}_i) > (\boldsymbol{\delta}'/b)^\top(\boldsymbol{\ell}_i - \boldsymbol{\ell}_t) \,, \qquad (2.4.3)$$

$$\mathbf{p}^\top(\boldsymbol{\ell}_t - \boldsymbol{\ell}_j) > (\boldsymbol{\delta}'/b)^\top(\boldsymbol{\ell}_t - \boldsymbol{\ell}_j) \,, \qquad (2.4.4)$$

and define $\boldsymbol{\delta}'' := \boldsymbol{\delta}'/b$. Now, $\widetilde{\mathbf{U}}\boldsymbol{\delta}'' = 0$ and $(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta}'' \neq 0$. Define $\mathbf{p}_1 := \mathbf{p} + \boldsymbol{\delta}''$ and $\mathbf{p}_2 := \mathbf{p} - \boldsymbol{\delta}''$. We can see that $p_{1r} \geq 0$ and $p_{2r} \geq 0$ $\forall r \in [n]$. Also,

$$\begin{aligned}\mathbf{1}^\top \mathbf{p}_1 &= \mathbf{1}^\top \mathbf{p} + \mathbf{1}^\top \boldsymbol{\delta}'' \\ &= 1 + 0 \,, \qquad\qquad \text{since } \widetilde{\mathbf{U}}\boldsymbol{\delta}'' = 0 \text{ and } \mathbf{1} \in \operatorname{col}(\widetilde{\mathbf{U}}^\top) \\ &= 1 \,.\end{aligned}$$

Similarly, $\mathbf{1}^\top \mathbf{p}_2 = 1$. Therefore, $\mathbf{p}_1$ and $\mathbf{p}_2$ are valid probability vectors in Δ_n.

Now, we claim that $\mathbf{p}_1 \in \mathcal{Q}^\ell_i$ and $\mathbf{p}_1 \notin \mathcal{Q}^\ell_t\ \forall t \neq i$. We have,

$$\begin{aligned}(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \mathbf{p}_1 &= (\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \mathbf{p} + (\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta}'' \\ &= 0 + (\boldsymbol{\ell}_i - \boldsymbol{\ell}_j)^\top \boldsymbol{\delta}''\,, \qquad \text{since } \mathbf{p} \in \mathcal{Q}^\ell_i \cap \mathcal{Q}^\ell_j \\ &< 0\,.\end{aligned}$$

This gives $\mathbf{p}_1 \notin \mathcal{Q}^\ell_j$. Moreover, $\forall t \neq i, j$, we have

$$\begin{aligned}(\boldsymbol{\ell}_i - \boldsymbol{\ell}_t)^\top \mathbf{p}_1 &= \mathbf{p}^\top(\boldsymbol{\ell}_i - \boldsymbol{\ell}_t) + \boldsymbol{\delta}''^\top(\boldsymbol{\ell}_i - \boldsymbol{\ell}_t) \\ &< 0\,, \qquad \text{by Eq. (2.4.3)}\ .\end{aligned}$$

Thus $\mathbf{p}_1 \in \mathcal{Q}^\ell_i$ and $\mathbf{p}_1 \notin \mathcal{Q}^\ell_t\ \forall t \neq i$. Similarly, $\mathbf{p}_2 \in \mathcal{Q}^\ell_j$ and $\mathbf{p}_2 \notin \mathcal{Q}^\ell_t\ \forall t \neq j$. Therefore, $\mathrm{Opt}(\ell, \mathbf{p}_1) \cap \mathrm{Opt}(\ell, \mathbf{p}_2) = \emptyset$. Moreover,

$$\begin{aligned}\mathbf{U}\mathbf{p}_1 &= \mathbf{U}\mathbf{p} + \mathbf{U}\boldsymbol{\delta}'' \\ &= \mathbf{U}\mathbf{p}\,, \qquad \text{since } \mathbf{U}\boldsymbol{\delta}' = 0 \\ &= \mathbf{U}\mathbf{p}_2\,.\end{aligned}$$

This gives us a contradiction since Γ will not be able to differentiate between $\mathbf{p}_1$ and $\mathbf{p}_2$, even though the optimal predictions for them with respect to ℓ are different; in particular, we get $\mathrm{pred}(\Gamma(\mathbf{p}_1)) = \mathrm{pred}(\mathbf{U}\mathbf{p}_1) = \mathrm{pred}(\mathbf{U}\mathbf{p}_2) = \mathrm{pred}(\Gamma(\mathbf{p}_2))$, and so we cannot have $\mathrm{pred}(\Gamma(\mathbf{p}_1)) \in \mathrm{Opt}(\ell, \mathbf{p}_1)$ and $\mathrm{pred}(\Gamma(\mathbf{p}_2)) \in \mathrm{Opt}(\ell, \mathbf{p}_2)$, i.e. (Γ, pred) cannot be ℓ-calibrated over Δ_n. Therefore we must have $d > \mathrm{affdim}(\mathbf{L}) - 1$. □

2.5 Calibrated Surrogates via Calibrated Nonlinear Properties

We now consider settings where one can exploit calibrated *nonlinear* properties to design convex calibrated surrogates in an even smaller number of dimensions than is possible via linear properties. We start by considering quantiles, which are 1-dimensional nonlinear (possibly interval-valued) properties; quantiles can be directly elicited via convex strictly proper scoring rules and lead to calibrated 1-dimensional surrogates for certain ordinal regression type losses (Section 2.5.1). We then develop a general framework for designing low-dimensional convex calibrated surrogates under 'low-noise' conditions by eliciting vectors of quantiles that yield 'coarse' information about a distribution (Section 2.5.2). We conclude with a result that gives a necessary condition for a general nonlinear property to be directly elicitable via a convex strictly proper scoring rule (Section 2.5.3).

2.5.1 Quantiles and Interval-Valued Properties

Quantiles and generalized quantiles have recently received significant attention in the property elicitation literature (Kiefer, 2010; Gneiting, 2011; Schervish et al., 2012; Grant and Gneiting, 2013; Steinwart et al., 2014). These are nonlinear properties; moreover, for discrete distributions, these properties can take a range of values over an interval. Therefore we will need to allow for interval-valued properties Γ that map each distribution $\mathbf{p} \in \Delta_n$ (or more generally, each $\mathbf{p} \in \mathcal{P}$ for some $\mathcal{P} \subseteq \Delta_n$) to a vector of *intervals*, $\Gamma(\mathbf{p}) \in \mathcal{I}^d$, where $\mathcal{I}$ denotes the set of all intervals on the real line. In this case, we will say a scoring rule $\psi : [n] \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ is proper for $\Gamma : \mathcal{P} \rightarrow \mathcal{I}^d$ if

$$\forall \mathbf{p} \in \mathcal{P} : \qquad \Gamma(\mathbf{p}) \subseteq \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^d} \mathbf{E}_{Y \sim \mathbf{p}}[\psi(Y, \mathbf{u})] \, ,$$

and strictly proper for Γ if the above holds with equality (i.e. no value $\mathbf{u} \notin \Gamma(\mathbf{p})$ is a minimizer).

Given a loss $\ell : [n] \times [k] \rightarrow \mathbb{R}_+$, we will say an interval-valued property $\Gamma : \mathcal{P} \rightarrow \mathcal{I}^d$ is ℓ-calibrated over $\mathcal{P}$ if $\exists$ pred : $\mathbb{R}^d \rightarrow [k]$ such that for all $\mathbf{p} \in \mathcal{P}$ and all convergent sequences $\{\mathbf{u}_m\}$ in $\mathbb{R}^d$,

$$\lim_{m \to \infty} \mathbf{u}_m \;\in\; \Gamma(\mathbf{p}) \;\;\Longrightarrow\;\; \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, \mathrm{pred}(\mathbf{u}_m)] \rightarrow \min_{t \in [k]} \mathbf{E}_{Y \sim \mathbf{p}}[\ell(Y, t)] \, .$$

Again, it can be shown that a strictly proper scoring rule ψ for an ℓ-calibrated interval-valued property $\Gamma : \mathcal{P} \rightarrow \mathcal{I}^d$ forms an ℓ-calibrated surrogate over $\mathcal{P}$.

Quantiles. For $\alpha \in (0,1)$, the α-quantile of $\mathbf{p} \in \Delta_n$ is defined as the interval

$$Q_\alpha(\mathbf{p}) = \big\{ u \in \mathbb{R} : \mathbf{P}_{Y \sim \mathbf{p}}(Y \leq u) \geq \alpha \text{ and } \mathbf{P}_{Y \sim \mathbf{p}}(Y \geq u) \geq 1 - \alpha \big\} \in \mathcal{I} \,. \tag{2.5.1}$$

It is known that the scoring rule $\psi : [n] \times \mathbb{R} \rightarrow \mathbb{R}_+$ defined as

$$\psi(y,u) = (1-\alpha) \cdot (u-y)_+ + \alpha \cdot (y-u)_+ \tag{2.5.2}$$

is a convex strictly proper scoring rule for the α-quantile, i.e. for the property $\Gamma : \Delta_n \rightarrow \mathcal{I}$ defined as $\Gamma(\mathbf{p}) = Q_\alpha(\mathbf{p})$. For the median $\Gamma(\mathbf{p}) = Q_{\frac{1}{2}}(\mathbf{p})$, the above scoring rule becomes $\psi(y,u) = \frac{1}{2}|u-y|$.

Example 2.5.1 (Generalized ordinal regression loss)**.** *Let $k = n$ and $\alpha \in (0,1)$, and consider the generalized ordinal regression loss $\ell : [n] \times [n] \rightarrow \mathbb{R}_+$ defined as*

$$\ell_{\mathrm{ord}(\alpha)}(y,t) = (1-\alpha)(t-y)_+ + \alpha(y-t)_+ \,.$$

It is easy to see that the α-quantile $\Gamma(\mathbf{p}) = Q_\alpha(\mathbf{p})$ is an $\ell_{\mathrm{ord}(\alpha)}$-calibrated nonlinear property over Δ_n; the scoring rule ψ in Eq. (2.5.2) is therefore a 1-dimensional convex calibrated surrogate for $\ell_{\mathrm{ord}(\alpha)}$ over Δ_n. Note that this is a significant improvement over what can be achieved with linear properties for these losses, e.g. for $\alpha = \frac{1}{2}$, the loss matrix $\mathbf{L}^{\mathrm{ord}(\alpha)}$ has affine dimension $n-1$, and thus by Theorem 2.4.4, any calibrated linear property for this loss must have dimension at least $n-2$.

2.5.2 Calibrated Surrogates under Low-Noise Conditions Using Vectors of Quantiles

We now give a general framework for constructing low-dimensional convex calibrated surrogates under suitable 'low-noise' conditions by eliciting a vector of quantiles that forms a calibrated nonlinear property under such conditions.

The broad idea is to estimate 'coarse' information about a distribution $\mathbf{p} \in \Delta_n$ using a vector of quantiles. Specifically, for any integer $s \in \mathbb{Z}_+$ ($s \geq 2$) and for a suitable set of distributions $\mathcal{P} \subseteq \Delta_n$, we define an $(s-1)$-dimensional interval-valued property $\Gamma_s : \mathcal{P} \rightarrow \mathcal{I}^{s-1}$ as follows:

$$\Gamma_s(\mathbf{p}) \;=\; Q_{\frac{1}{s}}(\mathbf{p}) \times \ldots \times Q_{\frac{s-1}{s}}(\mathbf{p}) \;\in\; \mathcal{I}^{s-1} \,. \tag{2.5.3}$$

From the discussion in Section 2.5.1, it follows that the scoring rule $\psi_s : [n] \times \mathbb{R}^{s-1} \rightarrow \mathbb{R}_+$ defined as

$$\psi_s(y, \mathbf{u}) = \sum_{i=1}^{s-1} \left(\left(1 - \frac{i}{s}\right) \cdot (u_i - y)_+ + \left(\frac{i}{s}\right) \cdot (y - u_i)_+ \right) \tag{2.5.4}$$

is a convex strictly proper scoring rule for Γ_s.

In order to design calibrated surrogates using the above vector-of-quantiles property Γ_s, we will find it convenient to define for each $y \in [n]$ a function $N_y : \mathbb{R}^{s-1} \rightarrow \mathbb{Z}_+$, which for each $\mathbf{u} \in \mathbb{R}^{s-1}$ counts how many times the label y appears in the vector $\lfloor \mathbf{u} \rfloor$ (where $\lfloor \mathbf{u} \rfloor = \big(\lfloor u_1 \rfloor, \ldots, \lfloor u_{s-1} \rfloor\big)^\top$):

$$N_y(\mathbf{u}) = \sum_{i=1}^{s-1} \mathbf{1}(y = \lfloor u_i \rfloor) \qquad \forall \mathbf{u} \in \mathbb{R}^{s-1} \,.$$

The following lemma shows that eliciting any $\mathbf{u} \in \Gamma_s(\mathbf{p})$ allows one to elicit for each $y \in [n]$ an interval of width at most $\frac{2}{s}$ containing p_y:

Lemma 2.5.2 (Vectors of quantiles give interval estimates for probabilities)**.** *Let $\mathcal{P} \subseteq \Delta_n$ and $\mathbf{p} \in \mathcal{P}$. Let $\Gamma_s : \mathcal{P} \rightarrow \mathcal{I}^{s-1}$ be defined as in Eq. (2.5.3) above, and let $\mathbf{u} \in \Gamma_s(\mathbf{p})$. Then for each $y \in [n]$, we have*

$$p_y \in \begin{cases} \left[\frac{N_y(\mathbf{u})-1}{s}, \frac{N_y(\mathbf{u})+1}{s}\right] & \textit{if } N_y(\mathbf{u}) \geq 1 \\[2ex] \left[0, \frac{1}{s}\right] & \textit{if } N_y(\mathbf{u}) = 0 \,. \end{cases}$$

Proof. Let $y \in [n]$. If $N_y(\mathbf{u}) = 0$, then no quantile in $\Gamma_s(\mathbf{p})$ consists of the singleton interval

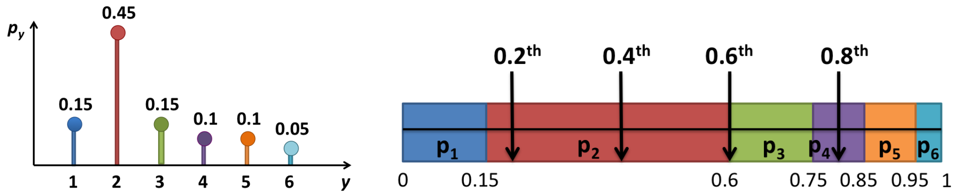


Figure 2: Illustration of quantile vector property $\Gamma_s(\mathbf{p})$ used to elicit coarse information about a distribution $\mathbf{p} \in \Delta_n$ (here $n = 6$, $s = 5$). See Example 2.5.3 for details.

$\{y\}$, and consequently, we must have $p_y \leq \frac{1}{s}$. Now suppose $N_y(\mathbf{u}) \geq 1$. Then the number of quantiles in $\Gamma_s(\mathbf{p})$ that consist of the singleton interval $\{y\}$ is at least $N_y(\mathbf{u}) - 2$ and at most $N_y(\mathbf{u})$, and therefore we must have $\frac{N_y(\mathbf{u})-1}{s} \leq p_y \leq \frac{N_y(\mathbf{u})+1}{s}$. □

Example 2.5.3 (Quantile vectors and probability interval estimates)**.** *Consider the example shown in Figure 2 ($n = 6$, $s = 5$). The figure shows the $\frac{1}{5}$, $\frac{2}{5}$, $\frac{3}{5}$ and $\frac{4}{5}$-quantiles of the probability vector $\mathbf{p} = (0.15, 0.45, 0.15, 0.1, 0.1, 0.05)^\top \in \Delta_6$. Here $Q_{\frac{1}{5}}(\mathbf{p}) = \{2\}$, $Q_{\frac{2}{5}}(\mathbf{p}) = \{2\}$, $Q_{\frac{3}{5}}(\mathbf{p}) = [2, 3]$, and $Q_{\frac{4}{5}}(\mathbf{p}) = \{4\}$, and so $\Gamma_5(\mathbf{p}) = \{2\} \times \{2\} \times [2, 3] \times \{4\}$. Consider $\mathbf{u} = (2,\ 2,\ 2.5,\ 4)^\top \in \Gamma_5(\mathbf{p})$. As can be seen, here $N_1(\mathbf{u}) = N_3(\mathbf{u}) = N_5(\mathbf{u}) = N_6(\mathbf{u}) = 0$; $N_2(\mathbf{u}) = 3$; and $N_4(\mathbf{u}) = 1$. Therefore by Lemma 2.5.2, we obtain the following interval estimates for elements of $\mathbf{p}$ from $\mathbf{u}$: p_1, p_3, p_5, $p_6 \in [0, 0.2]$; $p_2 \in [0.4, 0.8]$; and $p_4 \in [0, 0.4]$. Similarly, consider $\mathbf{u}' = (2,\ 2,\ 3,\ 4)^\top$, which also lies in $\Gamma_5(\mathbf{p})$. In this case, we would have $N_1(\mathbf{u}') = N_5(\mathbf{u}') = N_6(\mathbf{u}') = 0$; $N_2(\mathbf{u}') = 2$; and $N_3(\mathbf{u}') = N_4(\mathbf{u}') = 1$, and therefore we would get the following interval estimates for elements of $\mathbf{p}$ from $\mathbf{u}'$: p_1, p_5, $p_6 \in [0, 0.2]$; $p_2 \in [0.2, 0.6]$; and p_3, $p_4 \in [0, 0.4]$.*

Thus vectors of quantiles give coarse information about the probability distribution $\mathbf{p} \in \Delta_n$, and can be useful wherever it is sufficient to elicit not $\mathbf{p}$ exactly, but rather some intervals in which p_y lie. In particular, this can be useful for designing low-dimensional convex surrogates that are calibrated for a loss over a suitable set of 'low-noise' distributions. We give two such examples below, one for the multiclass 0-1 loss, and one for multiclass classification with a reject option.

Example 2.5.4. (O(log(n))-dimensional convex surrogate calibrated for 0-1 loss

under low-noise condition) *Let $k = n$ and consider the multiclass* 0-1 *loss $\ell_{\text{0-1}} : [n] \times [n] \to \mathbb{R}_+$ defined as*

$$\ell_{\text{0-1}}(y, t) = \mathbf{1}(y \neq t)\,.$$

Consider the following 'low-noise' condition, under which the highest-probability element is separated from the next highest-probability element by a probability difference of at least $\frac{2}{\lceil \log_2(n) \rceil}$:

$$\mathcal{P}^{\text{0-1}}_{\text{LN}} \;=\; \left\{ \mathbf{p} \in \Delta_n : \exists y \in [n] \textit{ such that } p_y > p_{y'} + \frac{2}{\lceil \log_2(n) \rceil} \;\forall y' \neq y \right\}.$$

Then it follows from Lemma 2.5.2 that for any $\mathbf{p} \in \mathcal{P}^{\text{0-1}}_{\text{LN}}$, by estimating a vector $\mathbf{u} \in \Gamma_{\lceil \log_2(n) \rceil}(\mathbf{p})$, one can accurately identify the largest-probability element under $\mathbf{p}$, $\text{argmax}_{y \in [n]}\, p_y$ *(and make an optimal prediction under $\ell_{\text{0-1}}$). Therefore the $(\lceil \log_2(n) \rceil - 1)$-dimensional property $\Gamma_{\lceil \log_2(n) \rceil}$ is $\ell_{\text{0-1}}$-calibrated over $\mathcal{P}^{\text{0-1}}_{\text{LN}}$ using* $\text{pred}^{\text{0-1}} : \mathbb{R}^{\lceil \log_2(n) \rceil - 1} \to [n]$ *satisfying*

$$\text{pred}^{\text{0-1}}(\mathbf{u}) \;\in\; \text{argmax}_{y \in [n]}\, N_y(\mathbf{u})\,.$$

For large n, for which the above low-noise condition is quite broad,[3] *this construction gives a significant improvement over the $n - 1$ dimensions needed for a convex surrogate to be calibrated for $\ell_{\text{0-1}}$ over Δ_n (Ramaswamy and Agarwal, 2012).*

Example 2.5.5. (O(log(n))-dimensional convex surrogate calibrated for multiclass classification with a reject option under low-noise condition) *Consider now a multiclass classification problem with a reject option. Here $k = n + 1$, with the prediction $(n + 1)$ corresponding to the 'reject' option; a common loss in this setting is the loss*

[3] Indeed, the low-noise condition $\mathcal{P}^{\text{0-1}}_{\text{LN}}$ here includes many probability distributions that are excluded from the commonly studied 'dominant-label' condition $\mathcal{P}^{\text{0-1}}_{\text{DL}} = \{\mathbf{p} \in \Delta_n : \max_{y \in [n]} p_y > \frac{1}{2}\}$, which is required for example for the common (n-dimensional) Crammer-Singer surrogate to be $\ell_{\text{0-1}}$-calibrated.

$\ell_{\text{reject}} : [n] \times [n+1] \to \mathbb{R}_+$ *defined as*

$$\ell_{\text{reject}}(y,t) = \begin{cases} \mathbf{1}(y \neq t) & \textit{if } t \in [n] \\ \frac{1}{2} & \textit{if } t = n+1. \end{cases}$$

Consider the following 'low-noise' condition, under which each probability element is separated from $\frac{1}{2}$ *by at least* $\frac{1}{\lceil \log_2(n) \rceil}$*:*

$$\mathcal{P}_{\text{LN}}^{\text{reject}} = \left\{ \mathbf{p} \in \Delta_n : p_y \notin \left[\frac{1}{2} - \frac{1}{\lceil \log_2(n) \rceil}, \frac{1}{2} + \frac{1}{\lceil \log_2(n) \rceil} \right] \forall y \in [n] \right\}.$$

Then it follows from Lemma 2.5.2 that for any $\mathbf{p} \in \mathcal{P}_{\text{LN}}^{\text{reject}}$*, by estimating a vector* $\mathbf{u} \in \Gamma_{\lceil \log_2(n) \rceil}(\mathbf{p})$*, one can accurately identify whether any label has probability greater than* $\frac{1}{2}$ *under* $\mathbf{p}$ *(and make an optimal prediction under* ℓ_{reject}*). Therefore the* $(\lceil \log_2(n) \rceil - 1)$*-dimensional property* $\Gamma_{\lceil \log_2(n) \rceil}$ *is* ℓ_{reject}*-calibrated over* $\mathcal{P}_{\text{LN}}^{\text{reject}}$ *using* $\text{pred}^{\text{reject}} : \mathbb{R}^{\lceil \log_2(n) \rceil - 1} \to [n]$ *defined as follows:*

$$\text{pred}^{\text{reject}}(\mathbf{u}) \;=\; \begin{cases} \text{argmax}_{y \in [n]} N_y(\mathbf{u}) & \textit{if } \exists y \in [n] \textit{ such that } N_y(\mathbf{u}) \geq \frac{\lceil \log_2(n) \rceil}{2} \\ n+1 & \textit{otherwise.} \end{cases}$$

To our knowledge, the above approach gives the first general framework for designing low-noise conditions together with convex surrogates that are calibrated under these conditions for different losses. In particular, the framework allows one to develop convex calibrated surrogates under any low-noise condition where a coarse estimate of the underlying probability vector suffices to make an optimal prediction under the loss of interest.

2.5.3 Necessary Condition for Convex Elicitability

As we have seen, linear properties and quantile-based properties are always directly elicitable by a convex strictly proper scoring rule. For general nonlinear properties, the following result gives a necessary condition for convex elicitability:

Theorem 2.5.1 (Necessary condition for convex elicitability of a property over Δ_n)**.** *Let $\Gamma : \Delta_n \to \mathbb{R}^d$. If Γ is directly elicitable via a convex proper scoring rule, then*

$$\dim(\Gamma^{-1}(\mathbf{u})) \;\geq\; n-d-1 \qquad \forall \mathbf{u} \in \Gamma(\mathrm{relint}(\Delta_n))\,.$$

Proof. Suppose Γ is directly elicitable via a convex proper scoring rule, and let $\psi : [n] \times \mathbb{R}^d \to \mathbb{R}_+$ be a convex strictly proper scoring rule for Γ. We will show that $\dim(\Gamma^{-1}(\mathbf{u})) \geq n-d-1 \;\forall \mathbf{u} \in \Gamma(\mathrm{relint}(\Delta_n))$.

Let $\mathbf{p} \in \mathrm{relint}(\Delta_n)$, and let $\mathbf{u}^* = \Gamma(\mathbf{p})$. Since ψ is strictly proper for Γ, we have

$$\mathbf{u}^* = \mathrm{argmin}_{\mathbf{u} \in \mathbb{R}^d}\, \mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u})]\,.$$

Moreover, since ψ is convex, we have

$$\mathbf{0} \in \partial(\mathbf{E}_{Y\sim\mathbf{p}}[\psi(Y,\mathbf{u}^*)]) = \sum_{y=1}^n p_y \partial\psi(y,\mathbf{u}^*)\,,$$

where $\partial\psi(y,\mathbf{u}^*)$ denotes the set of subdifferentials of $\psi(y,\mathbf{u})$ at $\mathbf{u}^*$ (if $\psi(y,\cdot)$ is differentiable, each such set is a singleton). Therefore for each $y \in [n]$, $\exists \mathbf{w}_y \in \partial\psi(y,\mathbf{u}^*)$ such that $\sum_{y=1}^n p_y \mathbf{w}_y = \mathbf{0}$. Let $\mathbf{A} = [\mathbf{w}_1 \cdots \mathbf{w}_n] \in \mathbb{R}^{d\times n}$, and let

$$\mathcal{H} = \{\mathbf{q} \in \Delta_n : \mathbf{A}\mathbf{q} = 0\} = \{\mathbf{q} \in \mathbb{R}^n : \mathbf{A}\mathbf{q} = \mathbf{0}, \mathbf{1}^\top\mathbf{q} = 1, -\mathbf{q} \leq \mathbf{0}\}\,,$$

where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector. We have $\mathbf{p} \in \mathcal{H}$, and also $-\mathbf{p} < \mathbf{0}$. Therefore, by Lemma 14 of Ramaswamy and Agarwal (2012), we have

$$\mu_{\mathcal{H}}(\mathbf{p}) \geq n - (d+1)\,,$$

where $\mu_{\mathcal{H}}(\mathbf{p})$ is the feasible subspace dimension of $\mathcal{H}$.[4] Now,

$$\begin{aligned}\mathbf{q} \in \mathcal{H} \implies \mathbf{A}\mathbf{q} = \mathbf{0} \quad &\implies \quad \mathbf{0} \in \sum_{y=1}^{n} q_y \partial\psi(y, \mathbf{u}^*) \\ &\implies \quad \mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}\in\mathbb{R}^d} \mathbf{E}_{Y\sim\mathbf{q}}[\psi(Y, \mathbf{u})] \\ &\implies \quad \Gamma(\mathbf{q}) = \mathbf{u}^*\,,\end{aligned}$$

which gives $\mathcal{H} \subseteq \Gamma^{-1}(\mathbf{u}^*)$, and therefore,

$$\dim(\Gamma^{-1}(\mathbf{u}^*)) \;\geq\; \mu_{\Gamma^{-1}(\mathbf{u}^*)}(\mathbf{p}) \;\geq\; \mu_{\mathcal{H}}(\mathbf{p}) \;\geq\; n - (d+1)\,.$$

Since $\mathbf{p} \in \operatorname{relint}(\Delta_n)$ was arbitrary, the result follows. □

Corollary 2.5.6. *Let $\Gamma : \Delta_n \to \mathbb{R}^d$ be d'-elicitable via a convex proper scoring rule in $d' \geq d$ dimensions. Then*

$$d' \;\geq\; n - \dim(\Gamma^{-1}(\mathbf{u})) - 1 \qquad \forall \mathbf{u} \in \Gamma(\operatorname{relint}(\Delta_n))\,.$$

[4] The feasible subspace dimension of a convex set $\mathcal{C}$ at $\mathbf{p} \in \mathcal{C}$ is defined as the dimension of the subspace $\mathcal{F}_{\mathcal{C}}(\mathbf{p}) \cup (-\mathcal{F}_{\mathcal{C}}(\mathbf{p}))$, where $\mathcal{F}_{\mathcal{C}}(\mathbf{p})$ is the cone of feasible directions of $\mathcal{C}$ at $\mathbf{p}$ (Ramaswamy and Agarwal, 2012).

Chapter 3

Information Elicitation in the Absence of Ground Truth

In the previous chapter we saw how tools from information elicitation can help the design of better surrogate losses for machine learning. In this chapter we will continue our discussion at the interface of machine learning and information elicitation, and see how information elicitation mechanisms in the absence of ground truth observations can benefit from using machine learning tools.

3.1 Introduction

3.1.1 Background

Recall from the previous chapter that truthful information elicitation mechanisms can be designed using proper scoring rules that take as input an agent's report and a ground truth observation from the underlying distribution. However, there are many applications where such ground truth observations are not available, for example, in massive open online courses (MOOCs) where the instructor does not grade student assignments but instead relies on students to grade each others assignments; in prediction markets where experts are asked about their opinion on future events; in surveys where respondents are asked about their feedback on a new product/feature. In the first example, there is an objective ground truth (instructor's grade) but it is costly to compute; in the second example, there is also an objective ground truth (outcome of the future event) but it is not known at the time of scoring; in the final example, there is no notion of an objective ground truth.

Peer prediction is a technique of eliciting truthful information in the absence of ground truth by comparing an agent's response with those of their peers. Peer prediction mechanisms leverage correlation in the reports of peers in order to score contributions. The main challenge of peer prediction is to incentivize agents to put effort to obtain a signal or form an opinion and then honestly report to the system. In recent years, peer prediction has been widely studied in several domains, including peer assessment in massively open online courses (MOOCs) (Shnayder and Parkes, 2016; Gao et al., 2016), for feedback on local places in a city (Mandal et al., 2016), and in the context of collaborative sensing platforms (Radanovic and Faltings, 2015d).

However, almost all general methods are essentially restricted to settings with homogeneous participants, whose signal distributions are identical. This is a poor fit with many suggested applications of peer prediction. Consider for example, the problem of peer assessment in MOOCs. DeBoer et al. (2013) and Wilkowski et al. (2014) observe that students differ based on their geographical locations, educational backgrounds, and level of commitment, and indeed the heterogeneity of assessment is clear from a study of Coursera data (Kulkarni et al., 2015). Simpson et al. (2013) observed that the users participating in a *citizen science* project can be categorized into five distinct groups based on their behavioral patterns in classifying an image as a Supernovae or not. A similar problem occurs in determining whether news headline is offensive or not. Depending on which social community a user belongs to, we should expect to get different opinions (Zafar et al., 2016). Moreover, Allcott and Gentzkow (2017) report that leading to the 2016 U.S. presidential election, people were more likely to believe the stories that favored their preferred candidate; Fourney et al. (2017) find that there is very low connectivity among Trump and Clinton supporters on social networks, which leads to confirmation bias among the two groups and clear heterogeneity about how they believe whether a piece of news is “fake” or not.

One obstacle to designing peer prediction mechanisms for heterogeneous agents is an impossibility result. No mechanism can provide strict incentives for truth-telling to a population of heterogeneous agents without knowledge of their signal distributions (Radanovic and Faltings, 2015c). This negative result holds for minimal mechanisms, which only elicit signals and not beliefs from agents. One way to alleviate this problem, without going to non-minimal mechanisms, is to use reports from the agents across multiple tasks to estimate their signal distributions. This is our goal: we want to design minimal peer prediction mechanisms for heterogeneous agents that use reports from the agents for both learning and scoring. We also want to provide robustness against coordinated misreports.

As a starting point, one can consider the *correlated agreement* (CA) mechanism proposed by Shnayder et al. (2016b). If the agents are homogeneous and the designer has knowledge of their joint signal distribution, the CA mechanism is *informed truthful*, i.e. no strategy profile, even if coordinated, can provide more expected payment than truth-telling, and the expected payment under an uninformed strategy (where an agent's report is independent of her signal) is strictly less than the expected payment under truth-telling. These two properties remove any incentive for coordinated deviations and strictly incentivize the agents to put effort in acquiring signals, respectively. In a detail-free variation, in which the designer learns the signal distribution from reports, approximate incentive alignment is provided (still maintaining the second property as a strict guarantee.) The detail-free CA mechanism can be extended to handle agent heterogeneity, but a naïve approach would require learning the joint signal distributions between every pair of agents, and the total number of reports that need to be collected would be prohibitive for many settings. Can we exploit machine learning techniques to address this requirement of learning joint signals for every pair of agents and to design a more efficient mechanism? In this chapter we seek to answer this question and design an efficient mechanism for heterogeneous agents.

3.1.2 Our Contributions

We design the first minimal and detail-free mechanism for peer prediction with heterogeneous agents, where the learning component has sample complexity that is only linear in the number of agents, while providing an incentive guarantee of approximate informed truthfulness. Like the CA mechanism, this is a multi-task mechanism in that each agent makes reports across multiple tasks. Our mechanism is robust to any coordination between agents as long as the task assignments are such that from an agent's perspective every other agent is equally likely to be her peer. Hence, our mechanism is robust to any coordination between agents that happens prior to task assignment. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned. For example, in the context of a MOOC, the organizer can anonymize the homeworks to be graded, and hence, it will require a lot of effort for students to figure out whose homeworks they are grading even after the homeworks have been assigned for grading. Since our mechanism has a learning component, the task assignments to agents should also be such that both the goals of incentive alignment and learning are simultaneously achieved. We consider two assignment schemes under which these goals can be achieved and analyze the sample complexity of our methods for these schemes.

The mechanism clusters the agents based on their reported behavior[1] and learns the pairwise correlations between these clusters. The clustering introduces one component of the incentive approximation, and could be problematic in the absence of a good clustering such that agents within a cluster behave similarly. Using eight real-world datasets, which contain reports of users on crowdsourcing platforms for multiple labeling tasks, we show that the clustering error is small in practice even when using a relatively small number of clusters. The second component of the incentive approximation stems from the need to learn the pairwise correlations between clusters; this component can be made arbitrarily small using a sufficient number of signal reports.

Another contribution of this chapter is to connect, we believe for the first time, the peer prediction literature with the extensive and influential literature on latent, confusion matrix models of label aggregation (Dawid and Skene, 1979b). The Dawid-Skene model assumes that signals are generated independently, conditional on a latent attribute of a task and according to an agent's confusion matrix. We cluster the agents based on their confusion matrices and then estimate the average confusion matrices within clusters using recent developments in tensor decomposition algorithms (Anandkumar et al., 2014; Zhang et al., 2016). These

[1] One could also consider clustering the agents based on their observable covariates as long as agents with similar covariates have similar 'signal type'. However, in the applications that we consider in this chapter, for example MOOCs, such covariates may not be observable, and hence, we only rely on agent reports for clustering.

average confusion matrices are then used to learn the pairwise correlations between clusters and design reward schemes to achieve approximate informed truthfulness.

In effect, the mechanism learns how to map one agent's signal reports onto the signal reports of the other agents. For example, consider the context of a MOOC, in which an agent in the "accurate" cluster accurately provides grades, an agent in the "extremal" cluster only uses grades 'A' and 'E', and an agent in the "contrarian" cluster flips good grades for bad grades and vice-versa. The mechanism might learn to positively score an 'A' report from an "extremal" agent matched with a 'B' report from an "accurate" agent, or matched with an 'E' report from a "contrarian" agent for the same essay. In practice, our mechanism will train on the data collected during a semester of peer assessment reports, and then cluster the students, estimate the pairwise signal distributions between clusters, and accordingly score the students (i.e., the scoring is done retroactively).

3.1.3 Related Work

We provide a brief review of the related work in peer prediction, and suggest (Faltings and Radanovic, 2017) for a detailed discussion. We focus our discussion on related work about minimal mechanisms, but remark that we are not aware of any non-minimal mechanisms (following from the work of Prelec (2004)) that handle agent heterogeneity. Miller et al. (2005) introduce the peer prediction problem, and proposed an incentive-aligned mechanism for the single-task setting. However, their mechanism requires knowledge of the joint signal distribution and is vulnerable to coordinated misreports. In regard to coordinated misreports, Jurca et al. (2009) show how to eliminate uninformative, pure-strategy equilibria through a three-peer mechanism, and Kong et al. (2016) provide a method to design robust, single-task, binary signal mechanisms (but need knowledge of the joint signal distribution). Frongillo and Witkowski (2017) provide a characterization of minimal (single task) peer prediction mechanisms.

Witkowski and Parkes (2013) introduce the combination of learning and peer prediction, coupling the estimation of the signal prior together with the shadowing mechanism. Some results make use of reports from a large population. Radanovic and Faltings (2015a), for example, establish robust incentive properties in a large-market limit where both the number of tasks and the number of agents assigned to each task grow without bound. Radanovic et al. (2016) provide complementary theoretical results, giving a mechanism in which truthfulness is the equilibrium with the highest payoff in the asymptote of a large population and with a structural property on the signal distribution.

Dasgupta and Ghosh (2013) show that robustness to coordinated misreports can be achieved for binary signals in a small population by using a multi-task mechanism. The idea is to reward agents if they provide the same signal on the same task, but punish them if one agent's report on one task is the same as another's on a different task. The Correlated Agreement (CA) mechanism (Shnayder et al., 2016b) generalizes this mechanism to handle multiple signals, and uses reports to estimate the correlation structure on pairs of signals without compromising incentives. In related work, Kong and Schoenebeck (2016, 2019) show that many peer prediction mechanisms can be derived within a single information-theoretic framework. Their results use different technical tools than those used by Shnayder et al. (2016b), and also include a different multi-signal generalization of the Dasgupta-Ghosh mechanism that provides robustness against coordinated misreports in the limit of a large number of tasks. Kong (2020) use this information-theoretic framework to design a mechanism that uses determinant based mutual information (DMI) to reward agents. This mechanism achieves *dominant truthfulness*, i.e. truthfulness dominates any other non-permutation strategy, using only a constant number of tasks. Shnayder et al. (2016c) adopt replicator dynamics as a model of population learning in peer prediction, and confirm that these multi-task mechanisms (including the mechanism by Kamble et al. (2015)) are successful at avoiding uninformed equilibria.

There are very few results on handling agent heterogeneity in peer prediction. For binary signals, the method of Dasgupta and Ghosh (2013) is likely to be an effective solution because their assumption on correlation structure will tend to hold for most reasonable models of heterogeneity. But it will break down for more than two signals, as explained by Shnayder et al. (2016b). Moreover, although the CA mechanism can in principle be extended to handle heterogeneity, it is not clear how the required statistical information about joint signal distributions can be efficiently learned and coupled with an analysis of approximate incentives. For a setting with binary signals and where each task has one of a fixed number of latent types, Kamble et al. (2015) design a mechanism that provides strict incentive compatibility for a suitably large number of heterogeneous agents, and when the number of tasks grows without bound (while allowing each agent to only provide reports on a bounded number of tasks). Their result is restricted to binary signals, and requires a strong regularity assumption on the generative model of signals. (Kong and Schoenebeck, 2016) design an information theoretic framework for peer prediction. Their mechanism pays each agent the mutual information between her report and her peer's report. This mechanism can be extended to the heterogeneous agents setting as long as we can measure the mutual information between all pairs of agents. However, such a mechanism would require the agents to provide reports on a large number of tasks.

Finally, we consider only binary effort of a user, i.e. the agent either invests effort and receives an informed signal or does not invest effort and receives an uninformed signal. Shnayder et al. (2016b) work with the binary effort setting and provide strict incentive for being truthful. Therefore, as long as the mechanism designer is aware of the cost of investing effort, the payments can be scaled to cover the cost of investing effort. The importance of motivating effort in the context of peer prediction has also been considered by Liu and Chen (2017b) and Witkowski et al. (2013).[2] See Mandal et al. (2016) for a setting with heterogeneous tasks but homogeneous agents. Liu and Chen (2017a) also designed single-task peer prediction mechanism for the same setting but only when each task is associated with a latent ground truth.

[2] Cai et al. (2015) work in a different model, showing how to achieve optimal statistical estimation from data provided by rational agents. They only focus on the cost of effort. They do not consider possible misreports, and thus their mechanism is also vulnerable to coordinated misreports.

3.1.4 Organization

In Section 3.2 we introduce the model for heterogeneous peer prediction. In Section 3.3 we present our mechanism and prove its truthfulness. In Section 3.4 we provide learning results for making our model detail-free. In Section 3.5 we present experiments on real-world data. We finally conclude in Section 3.6.

3.2 Model

Let notation $[t]$ denote $\{1, \ldots, t\}$ for $t \in \mathbb{N}$. We consider a population of agents $P = [\ell]$, and use indices such as p and q to refer to agents from this population. There is a set of tasks $M = [m]$. For example, a task can be either grading an essay or answering a question in an online rating sytem. When an agent performs a task, she receives a signal from $N = [n]$. Such a signal usually indicates the quality of the task i.e. the number of points assigned to the essay or how good the food is at a restaurant. The agents need to put some effort to get an informative signal about the task. As mentioned before, we assume that the effort of an agent is binary i.e. either the agent puts full effort and receives an informative signal or the agent puts no effort and receives a signal drawn uniformly at random. We also assume that the tasks are *ex ante* identical, that is, the signals of an agent for different tasks are sampled i.i.d. For example, in the essay grading scenario, if the essays assigned to any student are drawn uniformly at random from a large population of essays, the student's signal distribution for an assigned essay is *ex ante* almost identical to any other assigned essay.

Each agent is assigned a set of tasks and she decides, for each task, whether to put effort and receive an informative signal or put no effort and receive a random signal. This provides the agent with a set of signals, one for each task. Then the agent reports back to mechanism designer a set of signals, one for each assigned task. Before putting any effort to receive informative signals, the agents have no knowledge about the tasks apart from the fact they are ex-ante identical. Once the agents receive their signals, their reports are determined completely by these signals. In other words, the agents do not use any additional information to determine their reports. We will assume that, for each task, the message space and the signal space are the same. Since the payment made to the agents depend on their reported signals (*messages*), the reported signals can be very different than the observed signals. The goal of a peer prediction mechanism is to ensure that the agents put effort in all the tasks and report their signals truthfully. For the MOOC setting, a student spends some amount of time to figure out the grade of each of her assigned essays. She might also decide to not look at an essay and report an arbitrary grade. The goal of our mechanism is to ensure that the students put some effort to determine the grades of the essays and report them truthfully back to the platform. We work in the setting where the agents are heterogeneous, i.e., the distribution of signals can be different for different agents. These differences are captured by the agents' types and we say that the agents vary by *signal type.* In peer prediction, we compare the reports of an agent to the reports of their peers on the same tasks, and hence, we also need to talk about joint signal distribution of pairs of agents in addition to the signal distribution of an individual agent. In our setting, these joint signal distributions can be different for different pair of agents.

Let S_p, S_q denote random variables for the signal observed by agents p and q on some task. Let $D_{p,q}(i,j)$ denote the joint probability that agent p receives signal i while agent q receives signal j on a task, i.e. $D_{p,q}(i,j) = \Pr(S_p = i, S_q = j)$. Let $D_p(i)$ and $D_q(j)$ denote the corresponding marginal probabilities, i.e. $D_p(i) = \Pr(S_p = i)$ and , $D_q(j) = \Pr(S_q = j)$. An important part of our mechanisms are the *delta matrices* which are defined as follows. We define the *Delta matrix* $\Delta_{p,q}$ between agents p and q as

$$\Delta_{p,q}(i,j) = D_{p,q}(i,j) - D_p(i) \cdot D_q(j), \ \forall i,j \in [n]. \tag{3.2.1}$$

The delta matrices capture the correlation between pairs of realized signals. For example, if $\Delta_{p,q}(1,2) = D_{p,q}(1,2) - D_p(1)D_q(2) > 0$. This implies that $\Pr[S_p = 1 | S_q = 2] > \Pr[S_p = 1]$. Therefore, the event of agent p observing signal 1 is positively correlated with the event of agent q observing signal 2. This would also mean that the event that agent p receives signal

1 and agent q receives signal 2 is more likely when these signals are for the same task, than when they are for different tasks. Our mechanism will use these correlations to decide the score for an agent given the reports of the agent and her peers. The *correlated agreement* (CA) mechanism (Shnayder et al., 2016b) also uses these delta matrices to construct a scoring mechanism for agent reports, however, they work in a setting where agents are *exchangeable*, i.e. the delta matrix $\Delta_{p,q}$ is the same for all pairs p, q of agents.

Example 3.2.1. *For two agents p and q, consider the following joint signal distribution $D_{p,q}$ is*

$$D_{p,q} = \begin{bmatrix} 0.2 & 0.3 \\ 0.1 & 0.4 \end{bmatrix}$$

with marginal distributions $D_p = [0.5 \;\; 0.5]$ and $D_q = [0.3 \;\; 0.7]$, the Delta matrix $\Delta_{p,q}$ is

$$\Delta_{p,q} = \begin{bmatrix} 0.2 & 0.3 \\ 0.1 & 0.4 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.7 \end{bmatrix} = \begin{bmatrix} 0.05 & -0.05 \\ -0.05 & 0.05 \end{bmatrix}.$$

An agent's *strategy* defines, for every signal it may receive and each task it is assigned, a probability distribution over signals it will report. Shnayder et al. (2016b) show that it is without loss of generality for the class of mechanisms we study in this chapter to assume that an agent's strategy is uniform across different tasks. Hence, we will make the assumption that an agent's strategy is uniform across tasks. Formally, let R_p denote the random variable for the report of agent p for a given task. The strategy of agent p, denoted F^p, defines the distribution of reports for each possible signal i, with $F^p_{ir} = \Pr(R_p = r | S_p = i)$. Therefore if there are n signals then the strategy $F^p : [n] \to \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of all possible distributions with support in $[n]$. The collection of agent strategies, denoted $\{F^p\}_{p \in P}$, is the *strategy profile*. A strategy of agent p is *informed* if there exist distinct $i, j \in [n]$ and $r \in [n]$ such that $F^p_{ir} \neq F^p_{jr}$, i.e., if not all rows of F^p are identical. We say that a strategy is *uninformed* otherwise.

3.2.1 Multi-Task Peer Prediction

In this chapter we consider *multi-task peer prediction* mechanisms defined in Shnayder et al. (2016b), and extend them to the setting of heterogeneous agents. In these mechanisms, each agent performs multiple-tasks and the score of an agent depends on its reports and the reports of its peers. For each agent, a random subset of her tasks is designated as *bonus tasks*, and its complement is designated as *penalty tasks*, without the knowledge of the agent. These mechanisms are characterized by *scoring matrices* for each pair of agents, which are used to score agents' reports. In our mechanism, the scoring matrix $S_{p,q} : [n] \times [n] \rightarrow \{0, 1\}$ for agent pair p and q will be such that $S_{p,q}(i, j) = 1$ when the event that agent p receives signal i is positively correlated with the event that agent q receives signal j on the same task, otherwise $S_{p,q}(i, j) = 0$. We will thus use the delta matrices (which will be learnt from agent reports) to design these scoring matrices.

For signals i and j, if $S_{p,q}(i, j) = 1$, then, for each bonus task of an agent p, we will add 1 to her score for reporting i when the report of its peer agent q on the same task is j, otherwise we will not add anything. Additionally, for each bonus task of agent p, we randomly select a penalty task and subtract some score her total score based on her report on the penalty task. For signals i and j, if $S_{p,q}(i, j) = 1$, then we will subtract 1 from her score for reporting i on the penalty task when the report of its peer agent q on a different task is j, otherwise we will not subtract anything. The penalty is included in the score in order to avoid 'uninformative equilibria' where agents agree to report the same signal on every task without investing effort in gathering the signals. The total score of an agent will be sum of all the scores over all bonus tasks calculated this way.

In our mechanism the score of an agent on a bonus task will be '+1' when its report is positively correlated with the report of its peer agent on the same task. The score of an agent on a penalty task will be '-1' when its report is positively correlated with the report of its peer on a different task. The intuition behind our mechanism is that when signals i and j

of agents p and q are correlated then it will be more likely that agents receive this pair of signals on tasks they share than on tasks they do not share. Hence, the overall score will be positive in expectation, when agents are truthful. Whenever the agents use any uninformed strategy then the event that 'the report of agent p is i and the report of agent q is j' is as likely to happen when they perform the same task as it is when they perform different tasks. Hence, the expected payment of any uninformed strategy will be zero. The *correlated agreement* (CA) mechanism (Shnayder et al., 2016b) also uses a scoring matrix for scoring agent. However, in their homogeneous setting only one scoring matrix is required because the delta matrices are the same for each pair of agents. In our heterogeneous setting we have to use different scoring matrices for different pairs of agents.

Formally, for agent p, we denote the set of her bonus tasks by M_1^p and the set of her penalty tasks by M_2^p. To calculate the payment to an agent p for a bonus task $t \in M_1^p$, we do the following:

1. Randomly select an agent $q \in P \setminus \{p\}$ such that $t \in M_1^q$, and the set $M_2^p \cup M_2^q$ has at least 2 distinct tasks, and call q the *peer* of p.
2. Pick tasks $t' \in M_2^p$ and $t'' \in M_2^q$ randomly such that $t' \neq t''$ (t' and t'' are the penalty tasks for agents p and q respectively)
3. Let the reports of agent p on tasks t and t' be r_p^t and $r_p^{t'}$, respectively and the reports of agent q on tasks t and t'' be r_q^t and $r_q^{t''}$ respectively.
4. The payment of agent p for task t is then $S_{p,q}(r_p^t, r_q^t) - S_{p,q}(r_p^{t'}, r_q^{t''})$.

The total payment to an agent is the sum of payments for the agent's bonus tasks.

3.2.2 Task Assignments

Since we work in the setting where agents perform multiple tasks, and hence, it is important to address how these tasks are assigned to agents. Our mechanism has two requirements from any task assignment–

1. From an agent's perspective, every other agent is equally likely to be her peer. This requires agents not to know each other's task assignments before deciding a strategy. For example, if agents of one 'type' are more likely to be peers with agents of another 'type' based on their task assignments, then they can coordinate amongst themselves to decide a more profitable strategy than truth-telling. Our mechanism will be robust to coordinations that happen before the task assignments. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned.

2. We should always be able to find a peer agent q for any agent p. Precisely, the tasks are assigned in a way that for every agent p we can find a peer agent q such that q has performed at least one bonus task that p has performed, and we have reports from p and q for two different tasks which are not the same as the bonus task.

In addition, our mechanism has a learning component, where we learn about the correlation between agents' signals, and also cluster agents into groups. Hence, in order to learn these quantities, we need to collect sufficient reports from each agent. This imposes some other requirements for the task assignment. In Section 3.4 we propose two task assignment schemes that a principal can use that satisfy all these requirement.

3.2.3 Expected Payments

The expected payment to agent p under strategy profile $\{F^q\}_{q\in P}$ for any bonus task performed by her, equal across all bonus tasks as the tasks are *ex ante* identical, is given as

$$\begin{aligned}
u_p(F^p, \{F^q\}_{q\neq p}) &= \frac{1}{\ell-1}\sum_{q\neq p}\Bigg\{\sum_{i,j} D_{p,q}(i,j)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q) \\
&\quad - \sum_i D_p(i)\sum_{r_p} F^p_{ir_p}\sum_j D_q(j)\sum_{r_q} F^q_{jr_q}S_{p,q}(r_p,r_q)\Bigg\} \\
&= \frac{1}{\ell-1}\sum_{q\neq p}\Bigg\{\sum_{i,j}\left(D_{p,q}(i,j)-D_p(i)D_q(j)\right)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q)\Bigg\} \\
&= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q) \qquad (3.2.2)
\end{aligned}$$

3.2.4 Informed Truthfulness

Following Shnayder et al. (2016b), we define the notion of approximate informed truthfulness for a multi-task peer prediction mechanism.

Definition 3.2.2. *(ε-informed truthfulness) We say that a multi-task peer prediction mechanism is ε-informed truthful, for some $\varepsilon \geqslant 0$, if and only if for every strategy profile $\{F^q\}_{q\in P}$ and every agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon$, where $\mathbb{I}$ is the truthful strategy, and $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > u_p(F^p_0, \{F^q\}_{q\neq p})$ where F^p_0 is an uninformed strategy.*

An ε-informed truthful mechanism ensures that every agent prefers (up to ε) the truthful strategy profile over any other strategy profile, and strictly prefers the truthful strategy profile over any uninformed strategy. Moreover, no coordinated strategy profile provides more expected utility than the truthful strategy profile (upto ε). For a small ε, this is responsive to the main concerns about incentives in peer prediction: a minimal opportunity for coordinated manipulations, and a strict incentive to invest effort in collecting and reporting an informative signal.[3]

[3] We do not model the cost of effort explicitly in this chapter, but a binary cost model (effort → signal,

3.2.5 Learning and Agent Clustering

Suppose that one knows $\Delta_{p,q}$ for every pair of agents, then one can calculate the scoring matrices $S_{p,q}$ according to these delta matrices and use these scoring matrices to score the agents. It is not hard to prove (see Lemma 3.3.4 for a proof) that such an extension of the CA mechanism will be informed truthful. However, we seek to design a detail-free mechanism where one does not have the knowledge of delta matrices, and one needs to learn them from agent reports. However, it would require $\Omega(\ell^2)$ samples to learn the delta matrices between every pair of agents, which will often be impractical. Rather, the number of reports in a practical mechanism should scale closer to linearly in the number of agents.

In response, we will assume that agents can be (approximately) clustered into a bounded number K of agent signal types, such that agents of the same type have similar signal distributions. Hence, a cluster of agents will be treated as a meta-agent, and we will work with signal distributions of these meta-agents. Formally, let $G_1, \ldots, G_K$ denote a partitioning of agents into K clusters. With a slight abuse of notation, we also use $G(p)$ to denote the cluster to which agent p belongs.

In order to reduce the sample complexity of our mechanism, we want that the clustering of agents to be such that for each pair p, q of agents, the signals of meta-agents (clusters) $G(p)$ and $G(q)$ are correlated in a similar manner as the signals of agents p and q. With this in mind, for $s, t \in [K]$, let us define the cluster Delta matrix between clusters G_s and G_t to be the average signal correlation taken over all pairs of agents $p \in G_s$ and $q \in G_t$, i.e.

$$\Delta_{G_s,G_t} = \begin{cases} \frac{1}{|G_s|\times|G_t|}\sum_{p\in G_s, q\in G_t}\Delta_{p,q} & \text{if } s \neq t \\ \frac{1}{|G_s|^2 - G_s}\sum_{p,q\in G_s, q\neq p}\Delta_{p,q} & \text{if } s = t \end{cases}.$$

Now, the clustering of agents should be such that for each pair of agents p, q, we should be able to use $\Delta_{G(p),G(q)}$ as a proxy for $\Delta_{p,q}$. This will allow use learn Delta matrices for every

no-effort → no signal) can be handled in a straightforward way. See Shnayder et al. (2016b).

cluster pair, instead of learning Delta matrices for every agent pair. This intuition results in the following definition of an ε_1-*accurate* clustering.

Definition 3.2.3. *We say that clustering $G_1, \dots, G_K$ is ε_1-*accurate*, for some $\varepsilon_1 \geqslant 0$, if for every pair of agents $p, q \in P$,*

$$\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1, \tag{3.2.3}$$

where $\Delta_{G(p),G(q)}$ is the cluster Delta matrix *between clusters $G(p)$ and $G(q)$.*

Example 3.2.4. *Let there be 4 agents p, q, r and s. Let the pairwise Delta matrices be the following*

$$\Delta_{p,q} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \Delta_{p,r} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \Delta_{p,s} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}$$
$$\Delta_{q,r} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}, \Delta_{q,s} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \Delta_{r,s} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}$$

In this example, agents p and q tend to agree with each other, while agents r and s tend to agree with each other while disagreeing with p and q. Let the clustering be G_1, G_2 where p, q belong to G_1 and r, s belong to G_2. Then the cluster Delta matrices are the following

$$\Delta_{G_1,G_1} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \Delta_{G_1,G_2} = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & -0.1 \end{bmatrix}, \Delta_{G_2,G_2} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.$$

It is easy to observe that G_1, G_2 is a 0.2-accurate clustering.

Our mechanism will use an estimate of $\Delta_{G(p),G(q)}$ (instead of $\Delta_{p,q}$) to define the scoring matrix $S_{p,q}$. Thus, the incentive approximation will directly depend on the accuracy of the clustering as well as how good the estimate of $\Delta_{G(p),G(q)}$ is.

There is an inverse relationship between the number of clusters K and the clustering accuracy

ε_1: the higher the K, the lower the ε_1. In the extreme, we can let every agent be a separate cluster ($K = \ell$), which results in $\varepsilon_1 = 0$. But a small number of clusters is essential for a reasonable sample complexity as we need to learn $O(K^2)$ cluster Delta matrices. For instance, in Example 3.2.4 we need to learn 3 Delta matrices with clustering, as opposed to 6 without clustering. In Section 3.4, we give a learning algorithm that can learn all the pairwise cluster Delta matrices with $\widetilde{O}(K)$ samples given a clustering of the agents. In Section 3.5, we show using real-world data that a reasonably small clustering error can be achieved with relatively few clusters.

3.3 Correlated Agreement for Heterogeneous Agents

In this section we define our Correlated Agreement for Heterogeneous Agents (CAHU) mechanism, presented as Algorithm 1. Our mechanism builds upon the multi-task Correlated Agreement (CA) mechanism of Shnayder et al. (2016b), which uses the correlation between signals of different agents to design a scoring matrix to score the agents. However, since we work in a heterogeneous setting we will need to design different scoring matrices for different pairs of agents, based on the different correlations between different pairs.

For intuition, consider the case when one has knowledge of the Delta matrices for all pairs of agents. In this case, in the multi-task peer prediction framework defined in Section 3.2.1, the scoring matrices $S_{p,q}$ can be defined such that $S_{p,q}(i,j) = 1$ when $\Delta_{p,q} > 0$, and $S_{p,q}(i,j) = 0$ otherwise. Such a mechanism will be 0-informed truthful, as we prove in Lemma 3.3.4.

However, in order to design a detail-free mechanism with low sample complexity, we will assume that we have a clustering of agents such that the average cluster Delta matrices can be used as a proxy for agent Delta matrices. Hence, our mechanism works with a clustering of agents, and uses the cluster Delta matrices to design scoring matrices for pairs of agents. Here, we will describe our mechanism when a clustering as well as estimates of cluster Delta matrices are given as inputs to the mechanism. In Section 3.4, we will see how one can learn such a clustering and estimates of Delta matrices from agents reports.

Specifically, CAHU takes as input a clustering $G_1, \ldots, G_K$ of agents. It also takes as input matrices $\{\overline{\Delta}_{G_s,G_t}\}_{s,t \in [K]}$ which are estimates of the cluster Delta matrices $\{\Delta_{G_s,G_t}\}_{s,t \in [K]}$ defined in Section 3.2.5. The scoring matrix $S_{p,q}$ for agent pair p and q is then defined such that $S_{p,q}(i,j) = 1$ when $\Delta_{G(p),G(q)} > 0$, and $S_{p,q}(i,j) = 0$ otherwise, where $G(p)$ and $G(q)$ denote the clusters that p and q belong to, respectively. The CAHU mechanism then calculates the reward of an agent according to the framework of multi-task peer prediction discussed in Section 3.2.1. This would means that an agent p gets a positive score whenever her report and her peer q's report on a bonus task is such that there is positive correlation between the corresponding signals of clusters $G(p)$ and $G(q)$. However, we also include a penalty when this happens on different tasks. The idea is that if the clustering is ε_1-accurate and the estimates of cluster Delta matrices are accurate, then the mechanism should retain its truthfulness properties. With this in mind, we define an $(\varepsilon_1, \varepsilon_2)$-accurate input to the algorithm as follows

Definition 3.3.1. *We say that a clustering $\{G_s\}_{s \in [K]}$ and the estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t \in [K]}$ are $(\varepsilon_1, \varepsilon_2)$-accurate if*

- $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1$ *for all agents $p, q \in P$, i.e., the clustering is ε_1-accurate, and*
- $\|\Delta_{G_s,G_t} - \overline{\Delta}_{G_s,G_t}\|_1 \leqslant \varepsilon_2$ *for all clusters $s, t \in [K]$, i.e., the cluster Delta matrix estimates are ε_2-accurate.*

An ε_1 clustering intuitively means that if we pick one agent from cluster G_s and another agent from cluster G_t then their signal correlation is determined by the pair of clusters upto an error ε_1 and is independent of the identities of the agents. On the other hand ε_2-accurate clustering simple means that we can estimate the cluster delta matrices upto an error ε_2. When we have a clustering and estimates of the delta matrices which are $(\varepsilon_1, \varepsilon_2)$-accurate, we prove that the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful. In Section 3.4, we present algorithms that can learn an ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices.

```
Algorithm 1 Mechanism CAHU
Input:
    A clustering $G_1, \ldots, G_K$ such that $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1$ for all $p, q \in P$;
    estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$ such that $\|\overline{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant \varepsilon_2$ for all $s, t \in [K]$; and
    for each agent $p \in P$, her bonus tasks $M_1^p$, penalty tasks $M_2^p$, and responses $\{r_b^p\}_{b\in M_1^p \cup M_2^p}$.
Method:
for every agent $p \in P$ do
    for every task $b \in M_1^p$ do                                        ▷ Reward response $r_b^p$
        $q \leftarrow$ uniformly at random conditioned on $b \in M_1^q \cup M_2^q$ and (either $|M_2^q| \geqslant 2, |M_2^p| \geqslant 2$ or $M_2^q \neq M_2^p$)        ▷ Peer agent
        Pick tasks $b' \in M_2^p$ and $b'' \in M_2^q$ randomly such that $b' \neq b''$        ▷ Penalty tasks
        $S_{p,q} \leftarrow \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})$ †
        Reward to agent $p$ for task $b$ is $S_{p,q}\left(r_b^p, r_b^q\right) - S_{p,q}\left(r_{b'}^p, r_{b''}^q\right)$
    end for
end for
```

†Sign(x) = 1 if $x > 0$, and 0 otherwise.

Throughout the rest of this section, we will use ε_1 to denote the clustering error and ε_2 to denote the learning error. We remark that the clustering error ε_1 is determined by the level of similarity present in agent signal-report behavior, as well as the number of clusters K used, whereas the learning error ε_2 depends on how many samples the learning algorithm sees.

3.3.1 Analysis of CAHU

In this section we will prove the incentive properties of the CAHU mechanism. We will first present an overview of the proof, before presenting it formally. Recall that the expected payment of an agent in this setting is the following:

$$u_p(F^p, \{F^q\}_{q\neq p}) = \frac{1}{\ell - 1} \sum_{q\neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q)\,.$$

One can think of the expected payment to an agent p to be the average over all other agents q, the expected payment when q is p's peer agents. The expected payment when q is p's peer agent is given by the quantity $\sum_{i,j} \Delta_{p,q}(i,j) \cdot \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q)$.

For intuition, let us only consider deterministic strategies in this discussion. Our proof covers general randomized strategies. For deterministic strategies we have that

$$\sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) = S_{p,q}(F^p_i, F^q_j)\,,$$

where F^p_i and F^q_j denote (deterministic) reports of agents p and q given signals i and j, respectively. In this case the expected payment for p when q is her peer is $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F^p_i, F^q_j)$. Suppose that $\Delta_{p,q}$ has positive diagonals, and negative non-diagonals, and the scoring matrix $S_{p,q}$ is the identity matrix, then it is not hard to see that the maximum value of $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F^p_i, F^q_j)$ for any deterministic F^p and F^q is the trace of the matrix $\Delta_{p,q}$. Moreover, this maximum is achieved when F^p and F^q are truthful. Also, suppose that agents p and q adopt an uniformed strategy, say reporting '1' for every task, then the expected payment is $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(1,1)$ which is zero since the sum of the entries of the Delta matrices is always zero. For the general case, we will show that the maximum expected payment to p when agent q is her peer is given by $\sum_{i,j} \Delta_{p,q}(i,j) \cdot \mathrm{Sign}(\Delta_{p,q}(i,j))$. Hence, when $S_{p,q} = \mathrm{Sign}(\Delta_{p,q}(i,j))$, then this maximum is achieved when the agents are truthful. Also, the payment of any uninformed strategy is 0. Since, this holds for any peer agent q, this would imply informed truthfulness of the mechanism where $S_{p,q} = \mathrm{Sign}(\Delta_{p,q}(i,j))$. A similar argument also follow for any mixed strategies. A formal proof is presented in Lemma 3.3.4, and is very similar to the proof of informed truthfulness of the CA mechanism (Shnayder et al., 2016b).

However, we use approximate cluster Delta matrices instead of agent Delta matrices, to design the scoring matrices. Hence, we need to additionally worry about the effect of approximations due to clustering and learning on the incentive properties of our mechanisms. We will show that even under these approximation a truthful strategy will attain an expected reward that is close to the maximum possible expected reward. Precisely, we will show that when the clustering is ε_1-accurate and the cluster Delta matrix estimates are ε_2-accurate then the expected reward of a truthful strategy is at most $(\varepsilon_1 + \varepsilon_2)$ away from the maximum reward under any strategy and scoring matrices. Also, the expected reward of any uninformed strategy will always be zero. This will imply that CAHU is $(\varepsilon_1 + \varepsilon_2)$-informed truthful.

We will first need the following technical lemmas before proceeding to the main proof.

Lemma 3.3.2. *For any matrix $\widehat{S} \in \{0,1\}^{n \times n}$, and any probability distributions $\psi \in \mathcal{P}_n$ and $\phi \in \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of all probability distributions over $[n]$, we have that*

$$0 \leqslant \sum_{r_1, r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2) \phi_{r_2} \leqslant 1\,.$$

Proof. The fact that $\sum_{r_1,r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2)\phi_{r_2} \geqslant 0$ follows easily from the fact that $\psi_{r_1} \geqslant 0$, $\phi_{r_2} \geqslant 0$ and $\widehat{S}(r_1, r_2) \geqslant 0$ for all r_1 and r_2. The other direction follows from the following.

$$\begin{aligned}
\sum_{r_1, r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2) \phi_{r_2} &= \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} \widehat{S}(r_1, r_2) \phi_{r_2} \\
&\leqslant \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} 1 \cdot \phi_{r_2} && (\widehat{S}(r_1, r_2) \leqslant 1) \\
&= \sum_{r_1 \in [n]} \psi_{r_1} \cdot 1 && (\textstyle\sum_{r_2 \in [n]} \phi_{r_2} = 1) \\
&= 1 && (\textstyle\sum_{r_1 \in [n]} \psi_{r_1} = 1)
\end{aligned}$$

□

We now prove another technical lemma which gives an upper bound on the maximum payoff to an agent p under any scoring matrix.

Lemma 3.3.3. *Let $\{\widehat{S}_{p,q}\}_{p,q \in P}$ be an arbitrary set of scoring matrices where $\widehat{S}_{p,q} \in \{0,1\}^{n \times n}$ denotes the score matrix for agent p and agent q. Then for every strategy profile $\{F^q\}_{q \in P}$ we have that*

$$\sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p, r_q) \leqslant \sum_{i,j : \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j)\,.$$

Proof. We have that

$$\sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q) = \sum_{(i,j):\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q) + \sum_{(i,j):\Delta_{p,q}(i,j)\leqslant 0} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q)\,. \tag{3.3.1}$$

Now we make two observations. Firstly,

$$\sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \geqslant \sum_{(i,j):\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q)\,,$$

which follows from Lemma 3.3.2 as $\sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q) \leqslant 1$. Secondly,

$$\sum_{(i,j):\Delta_{p,q}(i,j)\leqslant 0} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q) \leqslant 0\,,$$

which again follows from Lemma 3.3.2 as $\sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p,r_q) \geqslant 0$.

Now, the desired bound follows from Equation 3.3.1 and the two observations above. □

We will now analyze our mechanism formally using the above lemmas. The derivation of the following result closely follows a similar analysis due to Shnayder et al. (2016b). We use $u^*_p(\cdot)$ to denote the utility of agent p when the scoring matrices are $\text{Sign}(\Delta_{p,q}(i,j))$, for all pairs p,q.

Lemma 3.3.4. *For a strategy profile $\{F^q\}_{q\in P}$ and an agent $p \in P$, define*

$$u^*_p(F^p, \{F^q\}_{q\neq p}) = \frac{1}{\ell-1} \sum_{q\neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S^*_{p,q}(r_p,r_q)$$

where $S^*_{p,q}(i,j) = \mathit{Sign}(\Delta_{p,q}(i,j))$ for all $i,j \in [n]$. Then, $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u^*_p(F^p, \{F^q\}_{q\neq p})$. Moreover, for any uninformed strategy F^p, $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > u^*_p(r, \{F^q\}_{q\neq p})$. This implies *informed-truthfulness of the mechanism where $S^*_{p,q}$ is used for scoring agents p and q.*

Proof. Let $\mathbf{1}[\cdot]$ denote the indicator function. Then the utility of the truthful strategy profile $\{\mathbb{I}, \{\mathbb{I}\}_{q \neq p}\}$ is given by

$$\begin{aligned}
u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) &= \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} \mathbf{1}[i = r_p] \cdot \mathbf{1}[j = r_q] \cdot S^*_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S^*_{p,q}(i,j) \\
&= \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j : \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j)
\end{aligned}$$

The utility of any other strategy profile $\{F^p, \{F^q\}_{q \neq p}\}$ is given by

$$u^*_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{i r_p} F^q_{j r_q} S^*_{p,q}(r_p, r_q) \, .$$

From Lemma 3.3.3 we then have

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) \geqslant u^*_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) \, .$$

For an uninformed strategy F^p such that all the rows of F^p are the same, i.e. $F^p_{i\cdot} = \psi$ for all i where ψ is a probability distribution, we have

$$\begin{aligned}
u^*_p(F^p, \{F^q\}_{q \neq p}) &= \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{i r_p} F^q_{j r_q} S^*_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} \psi_{r_p} F^q_{j r_q} S^*_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{j} \sum_{r_p, r_q} \psi_{r_p} F^q_{j r_q} S^*_{p,q}(r_p, r_q) \left(\sum_{i} \Delta_{p,q}(i,j) \right) = 0
\end{aligned}$$

The last equality follows since the row / column sum of delta matrices is zero. On the other hand, $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$, being a sum of only positive entries, is strictly greater than 0. □

We now prove our main theorem that $(\varepsilon_1 + \varepsilon_2)$-informed truthfulness holds when $(\varepsilon_1, \varepsilon_2)$-accurate clustering and learning holds.

Theorem 3.3.5. *With $(\varepsilon_1, \varepsilon_2)$-accurate clustering and learning, mechanism CAHU is $(\varepsilon_1 + \varepsilon_2)$-informed truthful if $\min_p u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \varepsilon_1 + \varepsilon_2$. In particular,*

1. *For every profile $\{F^q\}_{q \in P}$ and agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$.*

2. *For any uninformed strategy F^p_0, $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > u_p(F^p_0, \{F^q\}_{q \neq p})$.*

Proof. Fix a strategy profile $\{F^q\}_{q \in P}$. We first show that $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u_p(F^p, \{F^q\}_{q \neq p})$, and then show that $|u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})| \leqslant \varepsilon_1 + \varepsilon_2$. These together imply that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. For the former, we first observe (similarly, as in proof of Lemma 3.3.4) that the utility of truthful reporting when the scoring matrix $S^*_{p,q}(i,j) = \text{Sign}(\Delta_{p,q}(i,j))$, is given by

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j : \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j)$$

The utility $u_p(F^p, \{F^q\}_{q \in P \setminus \{p\}})$ of an agent p for any strategy profile $\{F^p, \{F^q\}_{q \in P \setminus \{p\}}\}$ under our mechanism, when the scoring matrix $S_{p,q} = \text{Sign}(\overline{\Delta}_{G(p),G(q)})$, is given by

$$u_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q)$$

Now, using Lemma 3.3.3 and the expressions for $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}})$ and $u_p(F^p, \{F^q\}_{q \in P \setminus \{p\}})$ we have that

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) \geqslant u_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) \,.$$

For the latter, we have

$$|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| = \left| \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j)\big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\big) \right| \tag{3.3.2}$$

$$\begin{aligned}
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} |\Delta_{p,q}(i,j)\big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\big)| \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} |\Delta_{p,q}(i,j) - \overline{\Delta}_{G(p),G(q)}(i,j)| \\
&= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \|\Delta_{p,q} - \overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 + \|\Delta_{G(p),G(q)} - \overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \varepsilon_1 + \varepsilon_2 = \varepsilon_1 + \varepsilon_2 .
\end{aligned}$$

To show that the third transition holds, we show that $|a \cdot (\mathrm{Sign}(a) - \mathrm{Sign}(b))| \leqslant |a - b|$ for all real numbers $a, b \in \mathbb{R}$. When $\mathrm{Sign}(a) = \mathrm{Sign}(b)$, this holds trivially. When $\mathrm{Sign}(a) \neq \mathrm{Sign}(b)$, note that the RHS becomes $|a| + |b|$, which is an upper bound on the LHS, which becomes $|a|$. The penultimate transition holds by ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices. This proves the first part of the theorem.

Now, we prove the second part of the theorem. For an uninformed strategy F^p such that all the rows of F^p are the same, i.e. $F_i^p = \psi$ for all i where ψ is a probability distribution, we have

$$\begin{aligned}
u_p(F^p, \{F^q\}_{q\neq p}) &= \frac{1}{\ell-1} \sum_{q\neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell-1} \sum_{q\neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} \psi_{r_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{j} \sum_{r_p,r_q} \psi_{r_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \left(\sum_i \Delta_{p,q}(i,j) \right) = 0\,,
\end{aligned}$$

where the last equality follows because the rows and columns of $\Delta_{p,q}$ sum to zero. Since $|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| \leqslant \varepsilon_1 + \varepsilon_2$ we have

$$u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2 > 0$$

as $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1 + \varepsilon_2$ for any p. □

The CAHU mechanism always ensures that there is no strategy profile which gives an expected utility more than $\varepsilon_1 + \varepsilon_2$ above truthful reporting. The condition $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1 + \varepsilon_2$ is required to ensure that any uninformed strategy gives strictly less than the truth-telling equilibrium. This is important to promote effort in collecting and reporting an informative signal. Note that, the learning error ε_2 can be made if we have sufficient amount of data. Therefore, we need to guarantee that $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1$ to ensure that any uninformed strategy gives strictly less than the truth-telling. Writing it out, this condition requires that for each agent p the following holds :

$$\frac{1}{\ell - 1} \sum_{q\neq p} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) > \varepsilon_1. \tag{3.3.3}$$

In particular, a sufficient condition for this property is that for every pair of agents the expected reward on a bonus task in the CA mechanism when making truthful reports is at least ε_1, i.e. for every pair of agents p and q,

$$\sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) > \varepsilon_1. \tag{3.3.4}$$

In turn, as pointed out by Shnayder et al. (2016b), the LHS in (3.3.4) quantity can be interpreted as a measure of how much positive correlation there is in the joint distribution on signals between a pair of agents. Note that it is not important that this is same-signal correlation. For example, this quantity would be large between an accurate and an alwayswrong agent in a binary-signal domain, since the positive correlation would be between one agent's report and the flipped report from the other agent.

The incentive properties of the mechanism are retained when used together with learning the cluster structure and cluster Delta matrices. However, we do assume that the agents do not reveal their task assignments to each other. If the agents were aware of the identities of the tasks they are assigned, they could coordinate on the task identifiers to arrive at a profitable coordinated strategy. This is reasonable in practical settings as the number of tasks is often large. The next theorem shows that even if the agents could set the scoring matrices to be an arbitrary function $\widehat{S}$ through any possible deviating strategies, it is still beneficial to use the scoring matrices estimated from the truthful strategies. Let $\widehat{S}$ be an arbitrary scoring function i.e. $\widehat{S}_{p,q}$ specifies the score matrix for two agents from p and q. We will write $\widehat{u}_p(F^p, \{F^q\}_{q\neq p})$ to denote the expected utility of agent p under the CAHU mechanism with the reward function $\widehat{S}$ and strategy profile $(F^p, \{F^q\}_{q\neq p})$.

Theorem 3.3.6. *Let $\{\widehat{S}_{p,q}\}_{p,q\in P}$ be an arbitrary set of scoring matrices where $\widehat{S}_{p,q} \in \{0,1\}^{n\times n}$ denotes the score matrix for agent p and agent q. Then for every profile $\{F^q\}_{q\in P}$ and agent $p \in P$, we have*

1. $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant \widehat{u}_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$.

2. *If* $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1$, *then for any uninformed strategy* F_0^p, $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \widehat{u}_p(F_0^p, \{F^q\}_{q\neq p})$.

Proof. Similar to the proof of Lemma 3.3.4, the utility of truthful reporting when the scoring matrix $S^*_{p,q}(i,j) = \text{Sign}(\Delta_{p,q}(i,j))$, is given by

$$u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\in P\setminus\{p\}}) = \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j)$$

The utility $\widehat{u}_p(F^p, \{F^q\}_{q\in P\setminus\{p\}})$ of an agent p for any strategy profile $\{F^p, \{F^q\}_{q\in P\setminus\{p\}}\}$

when the scoring matrix is $\widehat{S}_{p,q}$, is given by

$$\widehat{u}_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p, r_q)$$

Now, using Lemma 3.3.3 and the expressions for $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}})$ and $\widehat{u}_p(F^p, \{F^q\}_{q \in P \setminus \{p\}})$ we have that

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant \widehat{u}_p(F^p, \{F^q\}_{q \neq p}) \,.$$

Now the proof of Theorem 3.3.5 shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Using the result above we get $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant \widehat{u}_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Similar to the proof of Theorem 3.3.5 it can be shown that $\widehat{u}_p(F^p_0, \{F^q\}_{q \neq p}) = 0$ for any uninformed strategy F^p_0. The proof of Theorem 3.3.5 also shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$ can be made positive whenever $\min_p u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \varepsilon_1$. □

The above theorem implies that the incentive properties of our mechanism hold even when agents are allowed to coordinate their strategies and the mechanism is learned using reports from these coordinated strategies. To be precise, recall that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$ is the expected payment to agent p when the mechanism learns the true Delta matrix and the agent reports truthfully. This is no less than the expected payment minus $\varepsilon_1 + \varepsilon_2$ when the mechanism learns any other delta matrices and the agents misreport in any arbitrary way.

3.4 Learning the Agent Signal Types

In this section, we provide algorithms for learning a clustering of agent signal types from reports, and further, for learning the cluster pairwise Δ matrices. The estimates of the Δ matrices can then be used to give an approximate-informed truthful mechanism. Along the way, we couple our methods with the latent "confusion matrix" methods of Dawid and Skene (1979b).

Recall that m is the total number of tasks about which reports are collected. Reports on m_1 of these tasks will also be used for clustering, and reports on a further m_2 of these tasks will be used for learning the cluster pairwise Δ matrices. We consider two different schemes for assigning agents to tasks for the purpose of clustering and learning (see Figures 3 and 4):

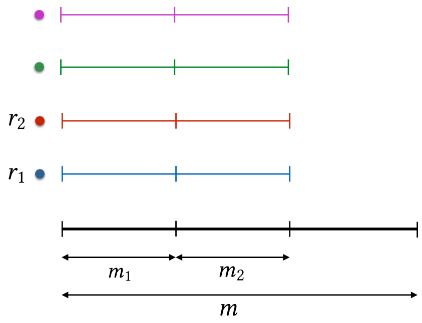


Figure 3: Fixed Task Assignment

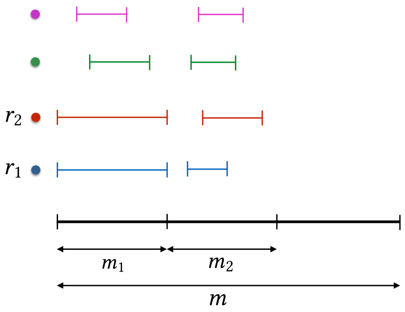


Figure 4: Uniform Task Assignment

1. **Fixed Task Assignment**: Each agent is assigned to the same, random subset of tasks of size $m_1 + m_2$ of the given m tasks.

2. **Uniform Task Assignment**: For clustering, we select two agents r_1 and r_2, uniformly at random, to be *reference agents*. These agents are assigned to a subset of tasks of size $m_1 (< m)$. For all other agents, we then assign a required number of tasks, s_1, uniformly at random from the set of m_1 tasks. For learning the cluster pairwise Δ-matrices, we also assign one agent from each cluster to some subset of tasks of size s_2, selected uniformly at random from a second set of $m_2 (< m - m_1)$ tasks.

For each assignment scheme, the analysis establishes that there are enough agents who have done a sufficient number of joint tasks. Table 1 summarizes the sample complexity results, stating them under two different assumptions about the way in which signals are generated.

[†] For an arbitrary m_2, this bound is Km_2 as long as m_2 is $\Omega\left(n^7/(\varepsilon')^2\right)$

[‡] In the no assumption approach (resp. Dawid-Skene Model), ε' is the error in the estimation of the joint probability distribution (resp. aggregate confusion matrix).

	No Assumption	Dawid-Skene
Fixed Assignment	Clustering: $\widetilde{O}\left(\frac{\ell n^2}{\gamma^2}\right)$ Learning: $\widetilde{O}\left(\frac{Kn^2}{(\varepsilon')^2}\right)$	Clustering: $\widetilde{O}\left(\frac{\ell n^2}{\gamma^2}\right)$ Learning: $\widetilde{O}\left(\frac{\ell n^7}{(\varepsilon')^2}\right)$
Uniform Assignment	Clustering: $\widetilde{O}\left(\frac{\ell n^2}{\gamma^2} + m_1\right)$ Learning: $\widetilde{O}\left(K m_2^{7/8} \sqrt{\frac{n^2}{(\varepsilon')^2}}\right)$	Clustering: $\widetilde{O}\left(\frac{\ell n^2}{\gamma^2} + m_1\right)$ Learning: $\widetilde{O}\left(\frac{Kn^7}{(\varepsilon')^2}\right)^\dagger$

Table 1: Sample complexity for the CAHU mechanism. The rows indicate the assignment scheme and the columns indicate the modeling assumption. Here ℓ is the number of agents, n is the number of signals, ε' is a parameter that controls learning accuracy ‡ , γ is a clustering parameter, K is the number of clusters, and m_1 (resp. m_2) is the size of the set of tasks from which the tasks used for clustering (resp. learning) are sampled.

3.4.1 Clustering

We proceed by presenting and analyzing a simple clustering algorithm.

Definition 3.4.1. *A clustering $G_1, \ldots, G_K$ is ε-good if for some $\gamma > 0$*

$$G(q) = G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 \leqslant \varepsilon - 4\gamma \ \forall p \in [\ell] \setminus \{q, r\} \tag{3.4.1}$$

$$G(q) \neq G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 > \varepsilon \ \forall p \in [\ell] \setminus \{q, r\} \tag{3.4.2}$$

We first show that an ε-good clustering, if exists, must be unique.

Theorem 3.4.2. *Suppose there exist two clustering $\{G_j\}_{j\in[K]}$ and $\{T_i\}_{i\in[K']}$ that are ε-good. Then $K' = K$ and $G_j = T_{\pi(j)}$ for some permutation π over $[K]$.*

Proof. Suppose equations 3.4.1 and 3.4.2 hold with parameters γ_1 and γ_2 respectively for the clusterings $\{G_j\}_{j\in[K]}$ and $\{T_i\}_{i\in[K']}$. If possible, assume there exist T_i and G_j such that $T_i \setminus G_j \neq \emptyset$, $G_j \setminus T_i \neq \emptyset$ and $T_i \cap G_j \neq \emptyset$. Pick $s \in T_i \cap G_j$ and $r \in G_j \setminus T_i$. Then we must have, for any $p \notin \{q, s, r\}$,

1. $\|\Delta_{pr} - \Delta_{ps}\|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i\in[K']}$)

2. $\|\Delta_{pr} - \Delta_{ps}\|_1 \leqslant \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j\in[K]}$)

Algorithm 2 Clustering

```
Input: $\varepsilon, \gamma$ such that there exists an $\varepsilon$-good clustering with parameter $\gamma$.
Output: A clustering $\{\widehat{G}_t\}_{t=1}^{\widehat{K}}$
$\widehat{G} \leftarrow \emptyset$, $\widehat{K} \leftarrow 0$    ▷ $\widehat{G}$ is the list of clusters, $\widehat{K} = |\widehat{G}|$
Make a new cluster $\widehat{G}_1$ and add agent 1
Add $\widehat{G}_1$ to $\widehat{G}$, $\widehat{K} \leftarrow \widehat{K} + 1$
for $i = 2, \dots, \ell$ do
    for $t \in [\widehat{K}]$ do
        Pick an arbitrary agent $q_t \in \widehat{G}_t$
        Pick $p_t \in [l] \setminus \{i, q_t\}$ (Fixed) or $p_t \in \{r_1, r_2\} \setminus \{i, q_t\}$ (Uniform), such that $p_t$
    has at least $\Omega(\frac{n^2 \log(K\ell/\delta)}{\gamma^2})$ tasks in common with both $q_t$ and $i$
        Let $\bar{\Delta}_{p_t,q_t}$ be the empirical Delta matrix from reports of agents $p_t$ and $q_t$
        Let $\bar{\Delta}_{p_t,i}$ be the empirical Delta matrix from reports of agents $p_t$ and $i$
    end for
    if $\exists t \in [\widehat{K}] : \|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \varepsilon - 2\gamma$ then
        add $i$ to $\widehat{G}_t$ (with ties broken arbitrarily for $t$)
    else
        Make a new cluster $\widehat{G}_{\widehat{K}+1}$ and add agent $i$ to it
        Add $\widehat{G}_{\widehat{K}+1}$ to $\widehat{G}$, $\widehat{K} \leftarrow \widehat{K} + 1$
    end if
end for
```

This is a contradiction. Now suppose $K' > K$. Then there must exist T_i and T_k such that $T_i \cup T_k \subseteq G_j$ for some j. Pick $q \in T_i$ and $r \in T_k$. Then, for any $p \notin \{q, r\}$

1. $\|\Delta_{pq} - \Delta_{pr}\|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i \in [K']}$)

2. $\|\Delta_{pq} - \Delta_{pr}\|_1 \leqslant \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j \in [K]}$)

This leads to a contradiction and proves that $K' \leqslant K$. Similarly we can prove $K \leqslant K'$. Therefore, we have shown that for each each G_j there exists i such that $G_j = T_i$. □

Since there is a unique ε-good clustering (up to a permutation), we will refer to this clustering as the *correct clustering*. The assumption that there exists an ε-good clustering is stronger than Equation (3.2.3) introduced earlier. In particular, identifying the correct clustering needs to satisfy Equation (3.4.2), i.e. the Δ-matrices of two agents belonging to two different clusters are different with respect to every other agent. So, we need low inter-cluster similarities in addition to high intra-cluster similarities. The pseudo-code for the clustering algorithm is

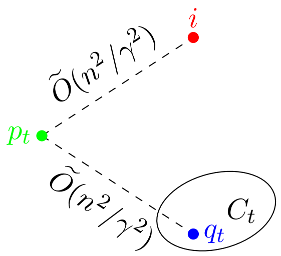


Figure 5: Algorithm 2 checks whether i and q_t are in the same cluster by estimating Δ_{p_t,q_t} and $\Delta_{p_t,i}$.

presented in Algorithm 2. This algorithm iterates over the agents, and forms clusters in a greedy manner. First, we prove that as long as we can find an agent p_t that has $\Omega\left(\frac{n^2\log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both q_t and i, then the clustering produced by Algorithm 2 is correct with probability at least $1-\delta$.

Theorem 3.4.3. *If for all $i \in P$ and $q_t \in G(i)$, there exists p_t which has $\Omega\left(\frac{n^2\log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both q_t and i, then Algorithm 2 recovers the correct clustering i.e. $\widehat{G}_t = G_t$ for $t = 1, \ldots, K$ with probability at least $1-\delta$.*

We need two key technical lemmas to prove Theorem 3.4.3. The first lemma shows that in order to estimate $\Delta_{p,q}$ with an L1 distance of at most γ, it is sufficient to estimate the joint probability distribution $D_{p,q}$ with an L1 distance of at most $\gamma/3$. With this, we can estimate the delta matrices of agent pairs from the joint empirical distributions of their reports.

Lemma 3.4.4. *For all $p, q \in P$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \gamma/3 \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant \gamma$.*

Proof.

$$
\begin{aligned}
\|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 &= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - \bar{D}_p(i)\bar{D}_q(j) - (D_{p,q}(i,j) - D_p(i)D_q(j))\right| \\
&= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| \\
&\qquad + \sum_{i,j} \left|\bar{D}_p(i)\bar{D}_q(j) - \bar{D}_p(i)D_q(j) + \bar{D}_p(i)D_q(j) - D_p(i)D_q(j)\right| \\
&\leqslant \gamma/3 + \sum_i \bar{D}_p(i) \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_j D_q(j) \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| \\
&\leqslant \gamma,
\end{aligned}
$$

as required. □

The second lemma is about learning the empirical distributions of reports of pairs of agents. This can be proved using Theorems 3.1 and 2.2 from the work of Devroye and Lugosi (2012).

Lemma 3.4.5. *Any distribution over a finite domain Ω is learnable within a L1 distance of d with probability at least $1-\delta$, by observing $O\left(\frac{|\Omega|}{d^2}\log(1/\delta)\right)$ samples from the distribution.*

We can use the above lemma to show that the joint distributions of reports of agents can be learned to within an $L1$ distance γ with probability at least $1-\delta/K\ell$, by observing $O\left(\frac{n^2}{\gamma^2}\log(K\ell/\delta)\right)$ reports on joint tasks.

Corollary 3.4.6. *For any agent pair $p,q \in P$, the joint distribution of their reports $D_{p,q}$ is learnable within a L1 distance of γ using $O\left(\frac{n^2}{\gamma^2}\log(K\ell/\delta)\right)$ reports on joint tasks with probability at least $1-\delta/K\ell$.*

We are now ready to prove Theorem 3.4.3.

Proof of Theorem 3.4.3. The proof is by induction on the number of agents ℓ. Suppose all the agents up to and including $i-1$ have been clustered correctly. Consider the i-th agent and suppose i belongs to the cluster G_t. Suppose $\widehat{G}_t \neq \emptyset$. Then using the triangle inequality we have

$$\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 + \|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 + \|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\|_1$$

Since $q_t \in G_t$, we have $\|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 \leqslant \varepsilon/2 - 4\gamma$. Moreover, using lemma 3.4.4 and corollary 3.4.6 we have that, with probability at least $1-\delta/K\ell$, $\|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\| \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \varepsilon/2 - 2\gamma$. On the other hand pick any cluster G_s such that $s \neq t$ and $\widehat{G}_s \neq \emptyset$. Then

$$\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 \geqslant \|\Delta_{p_s,q_s} - \Delta_{p_s,i}\| - \|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 - \|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1$$

Since $i \notin G_s$ we have $\|\Delta_{p_s,q_s} - \Delta_{p_s,i}\|_1 > \varepsilon/2$. Again, with probability at least $1-\delta/K\ell$, we have $\|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1 \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 > \varepsilon/2 - 2\gamma$. This ensures that condition on line (11) is violated for all clusters $s \neq t$. If $\widehat{G}_t \neq \emptyset$ this condition is satisfied and agent i added to cluster $\widehat{G}_t$, otherwise the algorithm makes a new cluster with agent i. Now note that the algorithm makes a new cluster only when it sees an agent belonging to a new cluster. This implies that $\widehat{K} = K$. Taking a union bound over the K choices of q_s for the K clusters, we see that agent i is assigned to its correct cluster with probability at least $1-\delta/\ell$. Finally, taking a union bound over all the ℓ agents we get the desired result. □

Next we show how the assumption in regard to task overlap is satisfied under each assignment scheme, and characterize the sample complexity of learning the clusterings under each scheme. In the fixed assignment scheme, all the agents are assigned to the same set of $m_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks. Thus, for each agent pair q_t and i, any other agent in the population can act as p_t. The total number of tasks performed is $O\left(\frac{\ell n^2}{\gamma^2}\log(K\ell/\delta)\right)$.

In the uniform assignment scheme, we select two agents r_1 and r_2 uniformly at random to be reference agents, and assign these agents to each of $m_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks. For all other agents we then assign $s_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks uniformly at random from this set of m_1 tasks. If $m_1 = s_1$, then the uniform task assignment is the same as fixed task assignment. However, in applications (e.g., (Karger et al., 2011)), where one wants the task assignments to be more uniform across tasks, it will make sense to use a larger value of m_1. The reference agent r_1 can act as p_t for all agent pairs q_t and i other than r_1. Similarly, reference r_2 can act as p_t for all agent pairs q_t and i other than r_2. If $q_t = r_1$ and $i = r_2$ or $q_t = r_2$ and $i = r_1$, then any other agent can act as p_t. The total number of tasks performed is $\Omega(\frac{\ell n^2}{\gamma^2}\log(K\ell/\delta) + m_1)$, which is sufficient for the high probability result.

3.4.2 Learning the Cluster Pairwise Δ Matrices

We proceed now under the assumption that the agents are clustered into K groups, $G_1, \ldots, G_K$. Our goal is to estimate the cluster-pairwise delta matrices Δ_{G_s,G_t} as required by Algorithm 1. We estimate the Δ_{G_s,G_t} under two different settings: when we have no model of the signal distribution, and in the Dawid-Skene latent attribute model.

Algorithm 3 Learning-Δ-No-Assumption

```
for $t = 1, \ldots, K$ do
    Chose agent $q_t \in G_t$ arbitrarily.
end for
for each pair of clusters $G_s, G_t$ do
    Let $q_s$ and $q_t$ be the chosen agents for $G_s$ and $G_t$, respectively.
    Let $\bar{D}_{q_s,q_t}$ be the empirical estimate of $D_{q_s,q_t}$ such that $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \leqslant \varepsilon'$ with
    probability at least $1 - \delta/K^2$
    Let $\bar{\Delta}_{q_s,q_t}$ be the empirical Delta matrix computed using $\bar{D}_{q_s,q_t}$
    Set $\bar{\Delta}_{G_s,G_t} = \bar{\Delta}_{q_s,q_t}$
end for
```

3.4.2.1 Learning the Δ-Matrices with No Assumption

We first characterize the sample complexity of learning the Δ-matrices in the absence of any modeling assumptions. In order to estimate $\bar{\Delta}_{G_s,G_t}$, Algorithm 3 first picks agent q_s from cluster G_s, estimates $\bar{\Delta}_{q_s,q_t}$ and use this estimate in place of $\bar{\Delta}_{G_s,G_t}$. For the fixed assignment scheme, we assign the agents q_s to the same set of tasks of size $O\left(\frac{n^2}{(\varepsilon')^2}\log(K/\delta)\right)$. For the uniform assignment scheme, we assign the agents to subsets of tasks of an appropriate size among the pool of m_2 tasks.

Theorem 3.4.7. *Given an ε-good clustering $\{G_s\}_{s=1}^K$, if the number of shared tasks between any pair of agents q_s, q_t is $O\left(\frac{n^2}{(\varepsilon')^2}\log(K/\delta)\right)$, then Algorithm 3 guarantees that for all s, t, $\|\bar{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$ with probability at least $1-\delta$. The total number of samples collected by the algorithm is $O\left(\frac{Kn^2}{(\varepsilon')^2}\log(K/\delta)\right)$ $\left(\text{resp. } O\left(Km_2^{7/8}\sqrt{\frac{n^2}{(\varepsilon')^2}\log(K/\delta)}\right) \text{ w.h.p.}\right)$ under the fixed (resp. uniform) assignment scheme.*

We first prove a sequence of lemmas that will be used to prove the result.

Lemma 3.4.8. *For every pair of agents p, q, we have*

$$\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant 2 \cdot \max_{a,b,c \in P : G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1.$$

Proof. Let $\Delta_{p,G(q)} = \frac{1}{|G(q)|}\sum_{r\in G(q)} \Delta_{p,r}$, then using the property of clusters we have

$$\begin{aligned}
\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 &= \left\|\Delta_{p,q} - \frac{1}{|G(p)|\,|G(q)|}\textstyle\sum_{u\in G(p),v\in G(q)} \Delta_{u,v}\right\|_1 \\
&= \left\|\frac{1}{|G(p)|\,|G(q)|}\textstyle\sum_{u\in G(p),v\in G(q)} \left(\Delta_{p,q} - \Delta_{u,v}\right)\right\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} \|\Delta_{p,q} - \Delta_{u,v}\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} \|\Delta_{p,q} - \Delta_{u,q}\|_1 + \|\Delta_{u,q} - \Delta_{u,v}\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} 2 \max_{a,b,c\in P:G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1 \\
&= 2 \max_{a,b,c\in P:G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1,
\end{aligned}$$

as required. □

The next lemma characterizes the error made by Algorithm 3 in estimating the Δ_{G_s,G_t}-matrices.

Lemma 3.4.9. *For any two agents $p \in G_s$ and $q \in G_t$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$.*

Proof. Lemma 3.4.4 shows that $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant 3\varepsilon'$.

Now,

$$\|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 + \|\Delta_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon.$$

The last inequality uses Lemma 3.4.8 □

Proof. (Theorem 3.4.7) By Lemma 3.4.5, to estimate $D_{p,q}$ within a distance of ε' with probability at least $1 - \delta/K^2$, we need $O\left(\frac{n^2}{(\varepsilon')^2}\log(K^2/\delta)\right)$. By a union bound over the K^2 pairs of clusters we see that with probability at least $1 - \delta$, we have $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \leqslant \varepsilon'$. This proves the first part of the theorem. When the assignment scheme is fixed, we can assign all the same tasks to K agents $\{q_t\}_{t=1}^K$, and hence the total number of samples is multiplied by K.

On the other hand, under the uniform assignment scheme, suppose each agent $\{q_t\}_{t=1}^K$ is assigned to a subset of s_2 tasks selected uniformly at random from the pool of m_2 tasks. Now consider any two agents q_s and q_t. Let X_i be an indicator random variable which is 1 when $i \in [m_2]$ is included in tasks of q_s, and 0 otherwise. Also, let Y_i be a similar random variable for the tasks of q_t. Let $Z_i = X_i \times Y_i$. The probability that both agents are assigned to a particular task i, $\Pr(Z_i = 1) = (s_2/m_2)^2$. Therefore, the expected number of overlapping tasks among the two agents is $m_2 \cdot \left(\frac{s_2}{m_2}\right)^2 = \frac{s_2^2}{m_2}$, i.e. $\mathrm{E}\left[\sum_i Z_i\right] = \frac{s_2^2}{m_2}$. Now, we want to bound the deviations from this expectations. Let $R_j = \mathrm{E}\left[\sum_{i=1}^{m_2} Z_i | X_1, \cdots, X_j, Y_1, \cdots, Y_j\right]$, then R_j is a Doob martingale sequence for $\sum_{i=1}^{j} Z_i$. Also, it is easy to see that this martingale sequence is bounded by 1, i.e. $|R_{j+1} - R_j| \leqslant 1$. Therefore, we apply the Azuma-Hoeffding bound (Lemma 3.4.10) as

$$\Pr\left[\left|\sum_i Z_i\right| > \frac{s_2^2}{2m_2}\right] \leqslant 2\exp\left\{-\frac{s_2^4}{8m_2^3}\right\}.$$

Now substituting $s_2 = m_2^{7/8} \cdot L^{1/2}$ where $L = O\left(\frac{n^2}{(\varepsilon')^2}\log(K^2/\delta)\right)$, we get

$$\Pr\left[\sum_i Z_i < m_2^{3/4} L/2\right] \leqslant 2\exp\left\{-\sqrt{m_2}L^2\right\}.$$

Taking a union bound over K^2 pairs of agents, if each agent completes $m_2^{7/8} \cdot L^{1/2}$ tasks selected uniformly at random from the pool of m_2 tasks, then the probability that any pair of agents has number of shared tasks L is at least $1 - K^2\exp\{-\sqrt{m_2}L^2\}$, which is exponentially small in m_2. □

Lemma 3.4.10. *Suppose $X_n, n \geqslant 1$ is a martingale such that $X_0 = 0$ and $|X_i - X_{i-1}| \leqslant 1$ for each $1 \leqslant i \leqslant n$. Then for every $t > 0$*

$$\Pr\left[|X_n| > t\right] \leqslant 2\exp\left\{-t^2/2n\right\}$$

3.4.2.2 Learning the Δ-matrices Under the Dawid-Skene Model

In this section, we assume that the agents receive signals according to the Dawid and Skene (1979a) model. Here, each task has a latent attribute and each agent has a confusion matrix to parameterize its signal distribution conditioned on this latent value. Recall two notations from the introduction : $D_p(i)$ is the marginal probability of observing signal i for agent p and $D_{p,q}(i,j)$ is the joint probability that the agents p and q observe signals i and j respectively. Then the Dawid-Skene Model is formally defined as :

- Let $\{\pi_k\}_{k=1}^n$ denote the prior probability over n latent values.
- Agent p has *confusion matrix* $C^p \in \mathbb{R}^{n\times n}$, such that $C^p_{ij} = D_p(S_p = j | T = i)$ where T is the latent value. Given this, the joint signal distribution for a pair of agents p and q

is

$$D_{p,q}(S_p = i, S_q = j) = \sum_{k=1}^{n} \pi_k C^p_{ki} C^q_{kj}, \tag{3.4.3}$$

and the marginal signal distribution for agent p is

$$D_p(S_p = i) = \sum_{k=1}^{n} \pi_k C^p_{ki}. \tag{3.4.4}$$

For cluster G_t, we write $C^t = \frac{1}{|G_t|} \sum_{p \in G_t} C^p$ to denote the aggregate confusion matrix of G_t. As before, we assume that we are given an ε-good clustering, $G_1, \ldots, G_K$, of the agents. Our goal is to provide an estimate of the Δ_{G_s,G_t}-matrices.

Lemma 3.4.11 proves that in order to estimate Δ_{G_s,G_t} within an L1 distance of ε', it is enough to estimate the aggregate confusion matrices within an L1 distance of $\varepsilon'/4$. So in order to learn the pairwise delta matrices between clusters, we first ensure that for each cluster G_t, we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'/4$ with probability at least $1 - \delta/K$, and then use the following formula to compute the delta matrices:

$$\Delta_{G_s,G_t}(i,j) = \sum_{k=1}^{n} \pi_k \bar{C}^s_{ki} \bar{C}^t_{kj} - \sum_{k=1}^{n} \pi_k \bar{C}^s_{ki} \sum_{k=1}^{n} \pi_k \bar{C}^t_{kj} \tag{3.4.5}$$

Lemma 3.4.11. *Forall G_a, G_b, $\|\bar{C}^a - C^a\|_1 \leqslant \varepsilon'/4$ and $\|\bar{C}^b - C^b\|_1 \leqslant \varepsilon'/4 \Rightarrow \|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\| \leqslant \varepsilon'$.*

Proof.

$$\begin{aligned}
\Delta_{G_a,G_b}(i,j) &= \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a,q\in G_b}\Delta_{p,q}(i,j) = \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a,q\in G_b} D_{p,q}(i,j) - D_p(i)D_q(j)\\
&= \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a,q\in G_b}\sum_k \pi_k C^p_{ki}C^q_{kj} - \sum_k \pi_k C^p_{ki}\sum_k C^q_{kj}\\
&= \sum_k \pi_k \left(\frac{1}{|G_a|}\sum_{p\in G_a} C^p_{ki}\right)\left(\frac{1}{|G_b|}\sum_{q\in G_b} C^q_{kj}\right)\\
&- \sum_k \pi_k \left(\frac{1}{|G_a|}\sum_{p\in G_a} C^p_{ki}\right)\sum_k \pi_k \left(\frac{1}{|G_b|}\sum_{q\in G_b} C^q_{kj}\right)\\
&= \sum_k \pi_k C^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}
\end{aligned}$$

Now

$$\begin{aligned}
&\|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\|_1 = \sum_{i,j}\left|\bar{\Delta}_{G_a,G_b}(i,j) - \Delta_{G_a,G_b}(i,j)\right|\\
&= \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \left(\sum_k \pi_k C^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right)\right|\\
&\leqslant \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki}C^b_{kj}\right| + \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right|\\
&= \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}C^b_{kj}\right|\\
&+ \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right|\\
&= \sum_k \pi_k \sum_j \left|\bar{C}^b_{kj} - C^b_{kj}\right|\sum_i \bar{C}^a_{ki} + \sum_k \pi_k \sum_i \left|\bar{C}^a_{ki} - C^a_{ki}\right|\sum_j C^b_{kj}\\
&+ \sum_k \pi_k \sum_i \bar{C}^a_{ki}\sum_{k'} \pi_{k'}\sum_j \left|\bar{C}^b_{k'j} - C^b_{k'j}\right| + \sum_k \pi_k \sum_j C^b_{kj}\sum_{k'} \pi_{k'}\sum_i \left|\bar{C}^a_{k'i} - C^a_{k'i}\right|\\
&= 2\sum_k \pi_k \sum_j \left|\bar{C}^b_{kj} - C^b_{kj}\right| + 2\sum_k \pi_k \sum_i \left|\bar{C}^a_{ki} - C^b_{ki}\right|\\
&\leqslant 2\|\bar{C}^a - C^a\|_1 + 2\|\bar{C}^b - C^b\|_1 \leqslant 4 \times \varepsilon'/4 = \varepsilon'
\end{aligned}$$

□

We now turn to the estimation of the aggregate confusion matrix of each cluster. Let us assume for now that the agents are assigned to the tasks according to the uniform assignment scheme, i.e. agent p belonging to cluster G_a is assigned to a subset of B_a tasks selected uniformly at random from a pool of m_2 tasks. For cluster G_a, we choose $B_a = \frac{m_2}{|G_a|}\ln(\frac{m_2 K}{\beta})$. This implies:

1. For each $j \in [m_2]$, $\Pr\left[\text{agent } p \in G_a \text{ completes task } j\right] = \frac{\log(m_2 K/\beta)}{|G_a|}$, i.e. each agent p in G_a is equally likely to complete every task j.

2. $\Pr\left[\text{task } j \text{ is unlabeled by } G_a\right] = \left(1 - \frac{\log(m_2 K/\beta)}{|G_a|}\right)^{|G_a|} \leqslant \frac{\beta}{m_2 K}$. Taking a union bound over the m_2 tasks and K clusters, we get the probability that any task is unlabeled is at most β. Now if we choose $\beta = 1/\text{poly}(m_2)$, we observe that with probability at least $1 - 1/\text{poly}(m_2)$, each task j is labeled by some agent in each cluster when $B_a = \widetilde{O}(\frac{m_2}{|G_a|})$.

All that is left to do is to provide an algorithm and sample complexity for learning the aggregate confusion matrices. For this, we will use n dimensional unit vectors to denote the reports of the agents (recall that there are n possible signals). In particular agent p's report on task j, $r_{pj} \in \{0,1\}^n$. If p's report on task j is c, then the c-th coordinate of r_{pj} is 1 and all the other coordinates are 0. The expected value of agent p's report on the jth task is $\mathrm{E}\left[r_{pj}\right] = \sum_{k=1}^{n} \pi_k C_k^p$ The aggregated report for a cluster G_t is given as $R_{tj} = \frac{1}{|G_t|}\sum_{p \in G_t} r_{pj}$.

Suppose we want to estimate the aggregate confusion matrix C^1 of some cluster G_1. To do so, we first pick three clusters G_1, G_2 and G_3 and write down the corresponding cross moments. Let (a, b, c) be a permutation of the set $\{1, 2, 3\}$. We have:

$$E[R_{aj}] = \sum_k \pi_k C_k^a \tag{3.4.6}$$

$$E[R_{aj} \otimes R_{bj}] = \sum_k \pi_k C_k^a \otimes C_k^b \tag{3.4.7}$$

$$E[R_{aj} \otimes R_{bj} \otimes R_{cj}] = \sum_k \pi_k C_k^a \otimes C_k^b \otimes C_k^c \tag{3.4.8}$$

The cross moments are asymmetric, however using Theorem 3.6 in the work by Anandkumar et al. (2014), we can write the cross-moments in a symmetric form.

Lemma 3.4.12. *Assume that the vectors $\{C_1^t, \dots, C_n^t\}$ are linearly independent for each $t \in \{1, 2, 3\}$. For any permutation (a, b, c) of the set $\{1, 2, 3\}$ define*

$$\begin{aligned} R'_{aj} &= \mathrm{E}\left[R_{cj} \otimes R_{bj}\right] \left(\mathrm{E}\left[R_{aj} \otimes R_{bj}\right]\right)^{-1} R_{aj} \\ R'_{bj} &= \mathrm{E}\left[R_{cj} \otimes R_{aj}\right] \left(\mathrm{E}\left[R_{bj} \otimes R_{aj}\right]\right)^{-1} R_{bj} \\ M_2 &= \mathrm{E}\left[R'_{aj} \otimes R'_{bj}\right] \ \textit{and}\ M_3 = \mathrm{E}\left[R'_{aj} \otimes R'_{bj} \otimes R_{cj}\right] \end{aligned}$$

$$\textit{Then } M_2 = \sum_{k=1}^{n} \pi_k C_k^c \otimes C_k^c \quad \textit{and} \quad M_3 = \sum_{k=1}^{n} \pi_k C_k^c \otimes C_k^c \otimes C_k^c$$

We cannot compute the moments exactly, but rather estimate the moments from samples observed from different tasks. Furthermore, for a given task j, instead of exactly computing the aggregate label R_{gj}, we select one agent p uniformly at random from G_g and use agent p's report on task j as a proxy for R_{gj}. We will denote the corresponding report as $\widetilde{R}_{gj}$. The next lemma proves that the cross-moments of $\{\widetilde{R}_{gj}\}_{g=1}^K$ and $\{R_{gj}\}_{g=1}^K$ are the same.

Lemma 3.4.13. 1. *For any group G_a, $\mathrm{E}\left[\widetilde{R}_{aj}\right] = \mathrm{E}\left[R_{aj}\right]$*

2. *For any pair of groups G_a and G_b, $\mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj}\right] = \mathrm{E}\left[R_{aj} \otimes R_{bj}\right]$*

3. For any three groups G_a, G_b *and* G_c, $\mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj} \otimes \widetilde{R}_{cj}\right] = \mathrm{E}\left[R_{aj} \otimes R_{bj} \otimes R_{cj}\right]$

Proof.

1. First moments of $\{\widetilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\widetilde{R}_{aj}\right] = \frac{1}{|G_a|} \sum_{p \in G_a} \mathrm{E}\left[r_{pj}\right] = E[R_{aj}]$$

2. Second order cross-moments of $\{\widetilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj}\right] = \sum_k \pi_k \mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj} | y_j = k\right] = \sum_k \pi_k \mathrm{E}\left[\widetilde{R}_{aj} | y_j = k\right] \otimes \mathrm{E}\left[\widetilde{R}_{bj} | y_j = k\right]$$

$$= \sum_k \pi_k \left(\frac{1}{|G_a|} \sum_{p \in G_a} C_k^p\right) \otimes \left(\frac{1}{|G_b|} \sum_{q \in G_b} C_k^q\right) = \sum_k \pi_k C_k^a \otimes C_k^b = \mathrm{E}\left[R_{aj} \otimes R_{bj}\right]$$

3. Third order cross-moments of $\{\widetilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj} \otimes \widetilde{R}_{cj}\right] = \sum_k \pi_k \mathrm{E}\left[\widetilde{R}_{aj} \otimes \widetilde{R}_{bj} \otimes \widetilde{R}_{cj} | y_j = k\right]$$

$$= \sum_k \pi_k \mathrm{E}\left[\widetilde{R}_{aj} | y_j = k\right] \otimes \mathrm{E}\left[\widetilde{R}_{bj} | y_j = k\right] \otimes \mathrm{E}\left[\widetilde{R}_{cj} | y_j = k\right]$$

$$= \sum_k \pi_k \left(\frac{1}{|G_a|} \sum_{p \in G_a} C_k^p\right) \otimes \left(\frac{1}{|G_b|} \sum_{q \in G_b} C_k^q\right) \otimes \left(\frac{1}{|G_c|} \sum_{r \in G_c} C_k^r\right)$$

$$= \sum_k \pi_k C_k^a \otimes C_k^b \otimes C_k^c = \mathrm{E}\left[R_{aj} \otimes R_{bj} \otimes R_{cj}\right]$$

□

Algorithm 4 Estimating Aggregate Confusion Matrix

```
Input: $K$ clusters of agents $G_1, G_2, \ldots, G_K$ and the reports $\widetilde{R}_{gj} \in \{0,1\}^n$ for $j \in [m]$ and $g \in [K]$
Output: Estimate of the aggregate confusion matrices $\bar{C}^g$ for all $g \in [K]$
Partition the $K$ clusters into groups of three
for Each group of three clusters $\{g_a, g_b, g_c\}$ do
    for $(a,b,c) \in \{(g_b, g_c, g_a), (g_c, g_a, g_b), (g_a, g_b, g_c)\}$ do
        Compute the second and the third order moments $\widehat{M_2} \in \mathbb{R}^{n \times n}$, $\widehat{M_3} \in \mathbb{R}^{n \times n \times n}$. ▷ Compute $\bar{C}^g$ and $\bar{\Pi}$ by tensor decomposition
        Compute whitening matrix $\widehat{Q} \in \mathbb{R}^{n \times n}$ such that $\widehat{Q}^T \widehat{M_2} \widehat{Q} = I$
        Compute eigenvalue-eigenvector pairs $(\widehat{\alpha}_k, \widehat{v}_k)_{k=1}^n$ of the whitened tensor $\widehat{M_3}(\widehat{Q}, \widehat{Q}, \widehat{Q})$ by using the robust tensor power method
        Compute $\widehat{w}_k = \widehat{\alpha}_k^{-2}$ and $\widehat{\mu}_k = (\widehat{Q}^T)^{-1} \widehat{\alpha} \widehat{v}_k$
        For $k = 1, \ldots, n$ set the $k$-th column of $\bar{C}^c$ by some $\widehat{\mu}_k$ whose $k$-th coordinate has the greatest component, then set the $k$-th diagonal entry of $\bar{\Pi}$ by $\widehat{w}_k$
    end for
end for
```

The next set of equations show how to approximate the moments M_2 and M_3:

$$\widehat{R}'_{aj} = \left(\frac{1}{m_2}\sum_{j'=1}^{m_2} \widetilde{R}_{cj'} \otimes \widetilde{R}_{bj'}\right)\left(\frac{1}{m_2}\sum_{j'=1}^{m_2} \widetilde{R}_{aj'} \otimes \widetilde{R}_{bj'}\right)^{-1} \widetilde{R}_{aj} \tag{3.4.9}$$

$$\widehat{R}'_{bj} = \left(\frac{1}{m_2}\sum_{j'=1}^{m_2} \widetilde{R}_{cj'} \otimes \widetilde{R}_{aj'}\right)\left(\frac{1}{m_2}\sum_{j'=1}^{m_2} \widetilde{R}_{bj'} \otimes \widetilde{R}_{aj'}\right)^{-1} \widetilde{R}_{bj} \tag{3.4.10}$$

$$\widehat{M_2} = \frac{1}{m_2}\sum_{j'=1}^{m_2} \widehat{R}'_{aj'} \otimes \widehat{R}'_{bj'} \quad \text{and} \quad \widehat{M_3} = \frac{1}{m_2}\sum_{j'=1}^{m_2} \widehat{R}'_{aj'} \otimes \widehat{R}'_{bj'} \otimes \widetilde{R}_{cj'} \tag{3.4.11}$$

We use the tensor decomposition algorithm (4) on $\widehat{M_2}$ and $\widehat{M_3}$ to recover the aggregate confusion matrix $\bar{C}^c$ and $\bar{\Pi}$, where $\bar{\Pi}$ is a diagonal matrix whose k-th component is $\bar{\pi}_k$, an estimate of π_k. In order to analyze the sample complexity of Algorithm 4, we need to make some mild assumptions about the problem instance. For any two clusters G_a and G_b, define $S_{ab} = \mathrm{E}\left[R_{aj} \otimes R_{bj}\right] = \sum_{k=1}^n \pi_k C_k^a \otimes C_k^b$. We make the following assumptions:

1. There exists $\sigma_L > 0$ such that $\sigma_n(S_{ab}) \geqslant \sigma_L$ for each pair of clusters a and b, where $\sigma_n(M)$ is the n-th smallest eigenvalue of M.

2. $\kappa = \min_{t\in[k]} \min_{s\in[n]} \min_{r\neq s} \left\{C^t_{rr} - C^t_{rs}\right\} > 0$

The first assumption implies that the matrices S_{ab} are non-singular. The smallest eigenvalue of S_{ab} controls how many samples we need to approximate S_{ab} from its sample mean. The second assumption implies that within a group, the probability of assigning the correct label is always higher than the probability of assigning any incorrect label. Note that this assumption might be false for an individual confusion matrix. However, we are averaging over all the users within a cluster to get the cluster average confusion matrix and unless a large fraction of individuals within a cluster has the propensity to mislabel i.e. assign large probability on incorrect labels, this assumption is usually satisfied. The following theorem gives the number of tasks each agent needs to complete to get an ε'-estimate of the aggregate confusion matrices. We will use the following two lemmas due to Zhang et al. (2016).

Lemma 3.4.14. *For any $\widehat{\varepsilon} \leqslant \sigma_L/2$, the second and the third empirical moments are bounded as*

$$\max\{\|\widehat{M_2} - M_2\|_{op}, \|\widehat{M_3} - M_3\|_{op}\} \leqslant 31\widehat{\varepsilon}/\sigma_L^3$$

with probability at least $1-\delta$ where $\delta = 6\exp\left(-(\sqrt{m_2}\widehat{\varepsilon} - 1)^2\right) + n\exp\left(-(\sqrt{m_2/n}\widehat{\varepsilon} - 1)^2\right)$

Lemma 3.4.15. *For any $\widehat{\varepsilon} \leqslant \kappa/2$, if the empirical moments satisfy*

$$\max\{\|\widehat{M_2} - M_2\|_{op}, \|\widehat{M_3} - M_3\|_{op}\} \leqslant \widehat{\varepsilon}H$$

$$\text{for} \quad H := \min\left\{\frac{1}{2}, \frac{2\sigma_L^{3/2}}{15n(24\sigma_L^{-1} + 2\sqrt{2})}, \frac{\sigma_L^{3/2}}{4\sqrt{3/2}\sigma_L^{1/2} + 8n(24/\sigma_L + 2\sqrt{2})}\right\}$$

then $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\widehat{\varepsilon}$, $\|\bar{\Pi} - \Pi\|_{op} \leqslant \widehat{\varepsilon}$ with probability at least $1-\delta$ where δ is defined in Lemma 3.4.14

Zhang et al. (2016) prove Lemma 3.4.14 when $\widehat{M_2}$ is defined using the aggregate labels R_{gj}. However, this lemma holds even if one uses the labels $\widetilde{R}_{gj}$. The proof is similar if one uses Lemma 3.4.13. We now characterize the sample complexity of learning the aggregate confusion matrices.

Theorem 3.4.16. *For any $\varepsilon' \leqslant \min\left\{\frac{31}{\sigma_L^2}, \frac{\kappa}{2}\right\} n^2$ and $\delta > 0$, if the size of the universe of shared tasks m_2 is at least $O\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}} \log\left(\frac{nK}{\delta}\right)\right)$, then we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'$ for each cluster G_t. The total number of samples collected by Algorithm 4 is $\widetilde{O}\left(Km_2\right)$ under the uniform assignment scheme.*

Proof. Substituting $\widehat{\varepsilon} = \widehat{\varepsilon}_1 H\sigma_L^3/31$ in lemma 3.4.14 we get

$$\max\{\|\widehat{M}_2 - M_2\|_{op}, \|\widehat{M}_3 - M_3\|_{op}\} \leqslant \widehat{\varepsilon}_1 H$$

with probability at least $1 - (6+n)\exp\left(-\left(\frac{m_2^{1/2}\widehat{\varepsilon}_1 H\sigma_L^3}{31n^{1/2}} - 1\right)^2\right)$. This substitution requires $\widehat{\varepsilon}_1 H\sigma_L^3/31 \leqslant \sigma_L/2$. Since $H \leqslant 1/2$, it is sufficient to have

$$\widehat{\varepsilon}_1 \leqslant 31/\sigma_L^2 \tag{3.4.12}$$

Now using Lemma 3.4.15 we see that $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\widehat{\varepsilon}_1$ and $\|\bar{\Pi} - \Pi\|_{op} \leqslant \widehat{\varepsilon}_1$ with the above probability. It can be checked that $H \geqslant \frac{\sigma_L^{5/2}}{230n}$. This implies that the bounds hold with probability at least $1 - (6+n)\exp\left(-\left(\frac{m_2^{1/2}\sigma_L^{11/2}\widehat{\varepsilon}_1}{7130n^{3/2}} - 1\right)^2\right)$. The second substitution requires

$$\widehat{\varepsilon}_1 \leqslant \kappa/2 \tag{3.4.13}$$

Therefore to achieve a probability of at least $1 - \delta$ we need

$$m_2 \geqslant \frac{7130^2 n^3}{\widehat{\varepsilon}_1^2 \sigma_L^{11}} \left(1 + \sqrt{\log\left(\frac{6+n}{\delta}\right)}\right)^2$$

It is sufficient that

$$m_2 \geqslant \Omega\left(\frac{n^3}{\widehat{\varepsilon}_1^2 \sigma_L^{11}} \log\left(\frac{n}{\delta}\right)\right)$$

to ensure $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\widehat{\varepsilon}_1$. For each k, $\|\bar{C}_k^c - C_k\|_1 \leqslant \sqrt{n}\|\bar{C}_k^c - C_k\|_2 \leqslant \sqrt{n}\|\bar{C}^c -$

$C\|_{op} \leqslant n\widehat{\varepsilon}_1$. Substituting $\widehat{\varepsilon}_1 = \widehat{\varepsilon}'/n^2$, we get $\|\bar{C}^c - C\|_1 = \sum_{k=1}^{n} \|\bar{C}_k^c - C_k\|_1 \leqslant n^2\widehat{\varepsilon}_1 = \widehat{\varepsilon}'$ when $m_2 = \Omega\left(\frac{n^7}{(\widehat{\varepsilon}')^2\sigma_L^{11}} \log\left(\frac{n}{\delta}\right)\right)$. By a union bound the result holds for all the clusters simultaneously with probability at least $1 - \delta K$. Substituting δ/K instead of δ gives the bound on the number of samples. Substituting $\widehat{\varepsilon}' = \widehat{\varepsilon}_1/n^2$ in equations 3.4.12 and 3.4.13, we get the desired bound on $\widehat{\varepsilon}'$.

Now to compute the total number of samples collected by the algorithm, note that each agent in cluster G_a provides $\frac{m_2}{|G_a|} \log\left(\frac{Km_2}{\beta}\right)$ samples. Therefore, total number of samples collected from cluster G_a is $m_2 \log\left(\frac{Km_2}{\beta}\right)$ and the total number of samples collected over all the clusters is $Km_2 \log\left(\frac{Km_2}{\beta}\right)$. □

Discussion. If the algorithm chooses $m_2 = \widetilde{O}\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\right)$, then the total number of samples collected under the uniform assignment scheme is at most $\widetilde{O}\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\right)$. So far we have analyzed the Dawid-Skene model under the uniform assignment scheme. When the assignment scheme is fixed, the moments of R_{aj} and $\widetilde{R}_{aj}$ need not be the same. In this case we will have to run Algorithm 4 with respect to the actual aggregate labels $\{R_{gj}\}_{g=1}^{K}$. This requires collecting samples from every member of a cluster, leading to a sample complexity of $O\left(\frac{\ell n^7}{(\varepsilon')^2\sigma_L^{11}} \log\left(\frac{nK}{\delta}\right)\right)$

In order to estimate the confusion matrices, Zhang et al. (2016) require each agent to provide at least $O\left(n^5 \log((\ell + n)/\delta)/(\varepsilon')^2\right)$ samples. Our algorithm requires $O\left(n^7 \log(nK/\delta)/(\varepsilon')^2\right)$ samples from each cluster. The increase of n^2 in the sample complexity comes about because we are estimating the aggregate confusion matrices in L1 norm instead of the infinity norm. Moreover when the number of clusters is small ($K \ll \ell$), the number of samples required from each cluster does not grow with ℓ. This improvement is due to the fact that, unlike Zhang et al. (2016), we do not have to recover individual confusion matrices from the aggregate confusion matrices.

Note that the approach based on the work of Dawid and Skene (1979b), for the uniform assignment scheme, does not require all agents to provide reports on the same set of shared tasks. Rather, we need that for each group of three clusters (as partitioned by Algorithm 4 on line 1) and each task, there should exist one agent from those three clusters who completes the same task. In particular the reports for different tasks can be acquired from different agents within the same cluster. The assignment scheme makes sure that this property holds with high probability.

We now briefly compare the learning algorithms under the no-assumptions and model-based approach. When it is difficult to assign agents to the same tasks, and when the number of signals is small (which is often true in practice), the Dawid-Skene method has a strong advantage. Another advantage of the Dawid-Skene method is that the learning error ε' can be made arbitrarily small since each aggregate confusion matrix can be learned with arbitrary accuracy, whereas the true learning error of the no-assumption approach is at least 2ε (see Theorem 3.4.7), and depends on the problem instance.

3.5 Clustering Experiments

Our goal in this section is to empirically evaluate the incentive that an agent has to use a non-truthful strategy under the CAHU mechanism in real-world scenarios. Recall that this *incentive error* comes from two sources:

- The clustering error. This represents how "clusterable" the agents are. From theory, we have the upper bound $\varepsilon_1 = \max_{p,q\in[\ell]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$.
- The learning error. This represents how accurate our estimates for the cluster Delta matrices are. From theory, we have the upper bound $\varepsilon_2 = \max_{i,j\in[K]} \|\Delta_{G_i,G_j} - \overline{\Delta}_{G_i,G_j}\|_1$.

Given this, the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful (Theorem 3.3.5).

In our experiments, we focus solely on the clustering error due to two reasons. First, the available real-world datasets have little overlap between the tasks performed by different agents, making it harder for us to learn their true pairwise Δ-matrices up to a reasonable accuracy and evaluate the error in our estimation. Note that the overlap is only needed to be able to *evaluate* the learning error of our approach; under the Dawid-Skene model, we do not require any overlap when using our approach in practice.

More importantly, the clustering error and the learning error differ in a key sense. Even with the best possible clustering, the clustering error ε_1 cannot be made arbitrarily small with a fixed number of clusters because it depends on how close the signal distributions of the agents really are. In contrast, the learning error ε_2 of the no-assumption approach is $3\varepsilon' + 2\varepsilon_1$, (Theorem 3.4.7) from which the part that does not depend on clustering (ε') can be made arbitrarily small by simply acquiring a sufficient amount of data about agents' behavior. Similarly, the learning error ε_2 in the Dawid-Skene approach — which we use in this experiment — can be made arbitrarily small too (Theorem 3.4.16). Hence, given a sufficient amount of data from the agents, the total error would be dominated by the clustering error ε_1. In particular, we show that in practice even a relatively small number of clusters lead to a small clustering error.

We use eight real-world crowdsourcing datasets. Six of these datasets are from the *SQUARE* benchmark (Sheshadri and Lease, 2013), selected to ensure a sufficient density of worker labels across different latent attributes as well as the availability of latent attributes for sufficiently many tasks. In addition, we also use the Stanford *Dogs* dataset (Khosla et al., 2011) and the *Expressions* dataset (Mozafari et al., 2014, 2012). Below, we briefly describe the format of tasks, the number of agents ℓ, and the number of signals n for each dataset.[4]

- Adult: Rating websites for their appropriateness, $\ell = 269$, $n = 4$.
- BM: Sentiment analysis for tweets, $\ell = 83$, $n = 2$.
- CI: Assessing websites for copyright infringement, $\ell = 10$, $n = 3$.
- Dogs: Identifying species from images of dogs, $\ell = 109$, $n = 4$.

[4] We filter each dataset to remove tasks for which the latent attribute is unknown, and remove workers who only perform such tasks. ℓ is the number of agents that remain after filtering.

- Expressions: Classifying images of human faces by expression, $\ell = 27$, $n = 4$.
- HCB: Assessing relevance of web search results, $\ell = 766$, $n = 4$.
- SpamCF: Assessing whether response to a crowdsourcing task was spam, $\ell = 150$, $n = 2$.
- WB: Identifying whether the waterbird in the image is a duck, $\ell = 53$, $n = 2$.

Since all datasets specify the latent value of the tasks, we adopt the Dawid-Skene model and estimate the confusion matrices from the frequency with which each agent p reports each label j in the case of each latent attribute i.

We first use a clustering algorithm to cluster the estimated confusion matrices. Typical clustering algorithms take a distance metric over the space of data points and attempt to minimize the maximum cluster diameter, which is the maximum distance between any two data points in a cluster. In contrast, our objective function (Equation (3.5.1)) is a complex function of the underlying confusion matrices. We therefore compare two approaches:

1) In this approach, we cluster the confusion matrices using the standard k-means++ algorithm with the $L2$ norm distance (available in Matlab) and hope that resulting clustering leads to a small error.[5]

2) In the following lemma, we derive a distance metric over confusion matrices for which the maximum cluster diameter is provably an upper bound on the clustering error, and use k-means++ with this metric (implemented in Matlab).[6] Note that computing this metric requires knowledge of the prior over the latent attribute (e.g., in the WB dataset, this would require knowing the probability that a random image of a waterbird

[5] We use $L2$ norm rather than $L1$ norm because the standard k-means++ implementation uses as the centroid of a cluster the confusion matrix that minimizes the sum of distances from the confusion matrices of the agents in the cluster. For $L2$ norm, this amounts to averaging over the confusion matrices, which is precisely what we want. For $L1$ norm, this amounts to taking a pointwise median, which does not even result in a valid confusion matrix. Perhaps for this reason, we observe that using the $L1$ norm performs worse.

[6] For computing the centroid of a cluster, we still average over the confusion matrices of the agents in the cluster. Also, since the algorithm is no longer guaranteed to converge (indeed, we observe cycles), we restart the algorithm when a cycle is detected, at most 10 times.

is a duck), which can be estimated easily from a small amount of ground truth data.

Lemma 3.5.1. *For all agents p, q, r, we have* $\|\Delta_{p,q} - \Delta_{p,r}\|_1 \leqslant 2 \cdot \sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}|$.

Proof. We have

$$\begin{aligned}
&\|\Delta_{p,q} - \Delta_{p,r}\|_1 = \sum_{i,j} |\Delta_{p,q}(i,j) - \Delta_{p,r}(i,j)| \\
&= \sum_{i,j} |D_{p,q}(i,j) - D_p(i)D_q(j) - D_{p,r}(i,j) + D_p(i)D_r(j)| \\
&= \sum_{i,j} |D_{p,q}(i,j) - D_{p,r}(i,j) - D_p(i)(D_q(j) - D_r(j))| \\
&= \sum_{i,j} \left| \sum_k \pi_k C^p_{ki} C^q_{kj} - \sum_k \pi_k C^p_{ki} C^r_{kj} - \sum_k \pi_k C^p_{ki} \left(\sum_l \pi_l C^q_{lj} - \sum_l \pi_l C^r_{lj} \right) \right| \\
&= \sum_{i,j} \left| \sum_k \pi_k C^p_{ki} \left(C^q_{kj} - C^r_{kj} \right) - \sum_k \pi_k C^p_{ki} \left(\sum_l \pi_l \left(C^q_{lj} - C^r_{lj} \right) \right) \right| \\
&\leqslant \sum_j \sum_k \pi_k \left| C^q_{kj} - C^r_{kj} \right| \sum_i C^p_{ki} + \sum_j \sum_k \pi_k \sum_l \pi_l \left| C^q_{lj} - C^r_{lj} \right| \sum_i C^p_{ki} \\
&= \sum_j \sum_k \pi_k \left| C^q_{kj} - C^r_{kj} \right| + \sum_j \sum_k \pi_k \sum_l \pi_l \left| C^q_{lj} - C^r_{lj} \right| \qquad [\text{Using } \sum_i C^p_{ki} = 1] \\
&\leqslant \sum_k \pi_k \sum_j \left| C^q_{kj} - C^r_{kj} \right| + \sum_k \pi_k \sum_l \pi_l \sum_j \left| C^q_{lj} - C^r_{lj} \right| \\
&= \sum_k \pi_k \sum_j \left| C^q_{kj} - C^r_{kj} \right| + \sum_l \pi_l \sum_j \left| C^q_{lj} - C^r_{lj} \right| \quad [\text{Using } \sum_k \pi_k = 1] \\
&= 2 \cdot \sum_k \pi_k \sum_j \left| C^q_{kj} - C^r_{kj} \right|,
\end{aligned}$$

as required. □

Note that $\sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}| \leqslant \|C^q - C^r\|_1$ because $\sum_j \left| C^q_{lj} - C^r_{lj} \right| \leqslant \|C^q - C^r\|_1$. Lemma 3.5.1, along with Lemma 3.4.8, shows that the incentive error due to clustering is upper bounded by four times the maximum cluster diameter under our metric, which defines the distance between C^q and C^r as $\sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}|$.

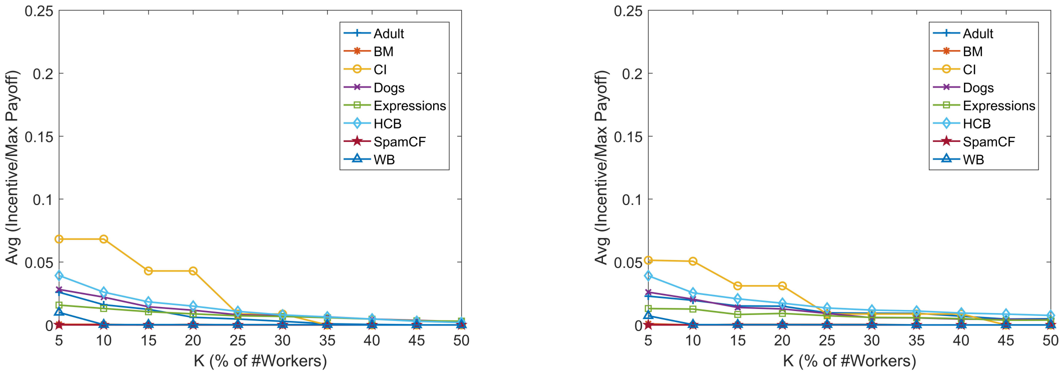
For each dataset, we vary the number of clusters K from 5% to 15% of the number of agents in the dataset. Within the k-means++ algorithm, we use 20 random seeds and choose the best clustering produced.

Next, we compute the clustering error. Instead of using the weak bound $\max_{p,q\in[\ell]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$ on the clustering error (which is nevertheless helpful for our theoretical results), we use the following tighter bound from the proof of Theorem 3.3.5.

$$|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| = \left| \frac{1}{(\ell-1)} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j)\big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\big) \right| \tag{3.5.1}$$

Assuming no learning error, this would be an upper bound on the incentive that agent p has to use a non-truthful strategy under the CAHU mechanism. We compare this bound to both the maximum payoff that agent p can receive and the expected payoff that agent p would receive under our mechanism, and plot the result averaged over p. Figures 6a and 6b similarly show the incentive of an average agent as a fraction of her *maximum* payoff with the standard $L2$ metric and with our custom metric, respectively. Figures 7a and 7b show the incentive of an average agent as a fraction of her *expected* payoff with standard $L2$ metric and with our custom metric, respectively. We note that the expected payoff is a stronger and more realistic benchmark than the maximum payoff.

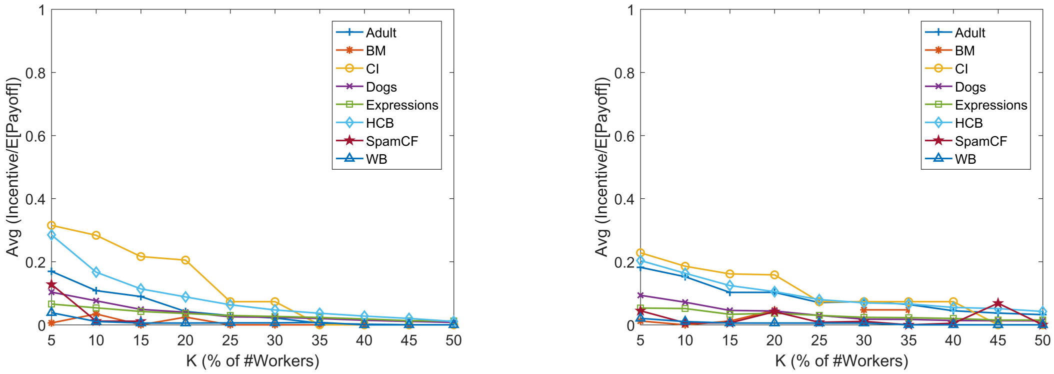
In comparison to both the maximum and the expected payoffs, the incentive error is small — less than 20% of the expected payoff and less than 5% of the maximum payoff — even with the number of clusters K as small as 15% of the number of workers. The number of agents does not seem to significantly affect this bound as long as the number of clusters is a fixed percentage of the number of agents. We also note that using our custom metric leads to a somewhat smaller error than using the standard $L2$ norm.



(a) Standard $L2$ metric

(b) Our custom metric

Figure 6: The incentive error as a fraction of the maximum payoff of an agent, averaged over agents, on 8 different data sets when using k-means++ with the $L2$ metric and with our custom metric



(a) Standard $L2$ metric

(b) Our custom metric

Figure 7: The incentive error as a fraction of the expected payoff of an agent, averaged over agents, on 8 different data sets when using k-means++ with the $L2$ metric and with our custom metric

3.6 Conclusion

We have provided the first, general solution to the problem of peer prediction with heterogeneous agents. This is a compelling research direction, where new theory and algorithms can help to guide practice. In particular, heterogeneity is likely to be quite ubiquitous due to differences in taste, context, judgment, and reliability across users. Beyond testing these methods in a real-world application such as marketing surveys, there remain interesting directions for ongoing research. For example, is it possible to solve this problem with a similar sample complexity but without a clustering approach? Is it possible to couple methods of peer prediction with optimal methods for inference in crowdsourced classification (Ok et al., 2016), and with methods for task assignment in budgeted settings (Karger et al., 2014)? This should include attention to adaptive assignment schemes (Khetan and Oh, 2016a) that leverage generalized Dawid-Skene models (Zhou et al., 2015), and could connect to the recent progress on task heterogeneity within peer prediction (Mandal et al., 2016). Finally, it is worth investigating if we can cluster the agents based on some observable characteristics like demographics, reputation scores etc and reduce the sample complexity of the original mechanism.

Chapter 4

Learning Multinomial Logit (MNL) Model from Choices

In this chapter we will begin our discussion at the interface of machine learning and discrete choice modeling. We present a fast and statistically efficient algorithm for learning the parameters of the multinomial logit choice model which is a widely studied model in discrete choice modeling.

4.1 Introduction

4.1.1 Background

Discrete choice modeling, which is studied in a variety of fields including economics and transportation, is concerned with the design of models of how humans make choices given a set of alternatives (Train, 2003; McFadden, 1974). These models have been used to explain or predict consumer choices in a wide range of applications. For example, in marketing these models are used for a variety of business problems such as pricing and product development; in transportation planning for estimating consumer demand for various transit choices; in labor economics for studying the participation in workforce and occupation choices; and so on. More recently, choice models have gained a lot of attention in machine learning due to the onset of online services in domains including entertainment and shopping, that use machine learning to recommend alternatives to users and help them make better choices. The presence of vast amount of consumer choice data in these applications makes it important to design efficient algorithms that can learn these models from data and use them in a variety of downstream applications such as demand estimation, product recommendation etc.

In this chapter we study the design of learning algorithms for the multinomial logit (MNL)/Plackett-Luce choice model which is one of the most popular models in discrete choice literature (Plackett, 1975; McFadden, 1974). Given a set of n items, the MNL model posits that there is a positive weight w_i associated with each item i, and the probability that item i

is chosen amongst all the items in a set S is $\frac{w_i}{\sum_{j \in S} w_j}$. The widely studied Bradley-Terry-Luce (BTL) model is a special case of the MNL model when the choice is pairwise, i.e. between two alternatives (Bradley and Terry, 1952a; Luce, 1959).

Learning choice models from pairwise choices has been an active area of research, and several algorithms have been proposed that are consistent under the BTL model (Negahban et al., 2017; Rajkumar and Agarwal, 2014; Hunter, 2004; Chen and Suh, 2015; Jang et al., 2016; Guiver and Snelson, 2009; Soufiani et al., 2013). The case of multiway choices has also received some attention recently (Maystre and Grossglauser, 2015; Jang et al., 2017; Chen et al., 2017b). Two popular algorithms are the rank centrality (RC) algorithm (Negahban et al., 2017) for the case of pairwise choices, and its generalization to the case of multiway choices, called the Luce spectral ranking (LSR) algorithm (Maystre and Grossglauser, 2015). The key idea behind these algorithms is to construct a *random walk* (equivalently a Markov chain) over the *comparison graph* on n items, where there is an edge between two items if they are compared in a pairwise or multiway choice set. This random walk is constructed such that its stationary distribution corresponds to the weights of the MNL/BTL model.

Given the widespread application of these algorithms, understanding their computational aspects is of paramount importance. For random walk based algorithms this amounts to analyzing the mixing/convergence time of their random walks to stationarity. In the case of rank centrality and Luce spectral ranking, ensuring that the stationary distribution of the random walk corresponds to the weights of the underlying model forces their construction to have self loops with large mass. These self loops can lead to a large mixing time of $\Omega\big(\xi^{-1} d_{\max}\big)$, where $d_{\max}$ is the maximum number of unique choice sets that any item participates in; and ξ is the spectral gap of the graph Laplacian. In practical settings $d_{\max}$ can be very large, for example when the graph follows a power-law distribution, and can even be $\Omega(n)$ if one item is compared to a large fraction of the items. In this chapter we seek to design faster algorithms for learning the MNL model whose running time has a mild or no dependence on $d_{\max}$.

4.1.2 Our Contributions

We show that it is possible to construct a faster mixing random walk whose mixing time is $O\big(\xi^{-1}\big)$. We are able to construct this random walk by relaxing the condition that its stationary distribution should exactly correspond to the weights of the MNL model, and instead imposing a weaker condition that the weights can be recovered through a linear transform of the stationary distribution. We call the resulting algorithm accelerated spectral ranking (ASR).

In addition to computational advantages, the faster mixing property of our random walk also comes with statistical advantages, as it is well understood that faster mixing Markov chains lend themselves to tighter perturbation error bounds (Mitrophanov, 2005). We are able to establish a sample complexity bound of $O\big(\xi^{-2}\, n\, \mathrm{poly}(\log n)\big)$, in terms of the *total variation* distance, for recovering the true weights under the MNL (and BTL) model for almost any comparison graph of practical interest. To our knowledge, these are the first sample complexity bounds for the general case of multiway choices under the MNL model. Negahban et al. (2017) show similar results in terms of L_2 error for the special case of BTL model. However, their bounds have an additional dependence on $d_{\max}$, due to the large mixing time of their random walk.

We also show that our algorithm can be viewed as a message passing algorithm. This connection provides a very attractive property to our algorithm – it can be implemented in a distributed manner with decentralized communication and choice data being stored in different machines.

We finally conduct several experiments on synthetic and real world datasets to compare the convergence time of our algorithm with the previous algorithms. These experiments confirm the behavior predicted by our theoretical analysis of mixing times– the convergence of our algorithm is in fact orders of magnitude faster than existing algorithms.

We summarize our contributions as follows:

1. **Faster Algorithm:** We present an algorithm for learning from pairwise choices under the BTL model, and more general multiway choices under the MNL model, that is provably faster than the previous algorithms of Negahban et al. (2017); Maystre and Grossglauser (2015). We also give experimental evidence supporting this fact.

2. **New and Improved Error Bounds:** We present the first error bounds for parameter recovery by spectral ranking algorithms under the general MNL model for any general (connected) comparison graph. These bounds improve upon the existing bounds of Negahban et al. (2017) for the special case of the BTL model.

3. **Message Passing Interpretation:** We provide an interpretation of our algorithm as a message passing/belief propagation algorithm. This connection can be used to design a decentralized distributed algorithm, which can work with distributed data storage.

4.1.3 Organization

In Section 4.2 we describe the problem formally. In Section 4.3 we present our algorithm for learning under the MNL/BTL model. In Section 4.4 we analyze the mixing time of our random walk, showing that our random walk converges much faster than existing approaches. In Section 4.5 we give bounds on sample complexity for recovery of MNL parameters with respect to the total variation distance. In Section 4.6 we give a message passing view of our algorithm. In Section 4.7 we provide experimental results on synthetic and real world datasets.

4.2 Problem Setting and Preliminaries

We consider a setting where there are n items, and one observes noisy pairwise or multiway choices between these items. We will assume that these choices are generated according to the multinomial logit (MNL) model, which posits that each item $i \in [n]$ is associated with a (unknown) weight/score $w_i > 0$, and the probability that item i is chosen is proportional to its weight w_i. More formally, when there is a (multiway) comparison between items of a set

$S \subseteq [n]$, for $i \in S$, we have

$$p_{i|S} := \Pr(i \text{ is chosen in } S) = \frac{w_i}{\sum_{j \in S} w_j}\,.$$

This model is also referred to as the Plackett-Luce model, and it reduces to the Bradley-Terry-Luce (BTL) model in the special case of pairwise choices, i.e. $|S| = 2$. Let $\mathbf{w} \in \mathbb{R}_+^n$ be the vector of weights, i.e. $\mathbf{w} = (w_1, \cdots, w_n)^\top$. Note that this model is invariant to any scaling of $\mathbf{w}$, so for uniqueness we will assume that $\sum_{i=1}^n w_i = 1$, i.e. $\mathbf{w} \in \Delta_n$ where Δ_n is the n-dimensional probability simplex.

The choice data is of the following form: there are d different *choice sets* $S_1, \cdots, S_d \subseteq [n]$, with $|S_a| = m$ for all $a \in [d]$ and some constant $m < n$. For each set S_a, for $a \in [d]$, one observes L independent m-way choices between items in S_a, drawn according to the MNL model. The assumptions that each choice set is of the same size m, and each set is compared an equal L number of times, are only for simplicity of exposition, and we give a generalization in the Appendix. We will denote by y_a^l the l-th choice amongst items of S_a, for $l \in [L]$ and $a \in [d]$.

Given choice data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^d$, where $\mathbf{y}_a = (y_a^1, \cdots, y_a^L)$, the problem is to find a weight vector $\widehat{\mathbf{w}} \in \Delta_n$, which is close to the true weight vector $\mathbf{w}$ under some notion of error/distance. More formally, the problem is to find $\widehat{\mathbf{w}} \in \Delta_n$, such that $\|\widehat{\mathbf{w}} - \mathbf{w}\|$ can be bounded in terms of the parameters n, L, and m, for some norm $\|\cdot\|$. We will give results in terms of the total variation distance, which for two vectors $\mathbf{u}, \widehat{\mathbf{u}} \in \Delta_n$ is defined as

$$\|\mathbf{u} - \widehat{\mathbf{u}}\|_{\mathrm{TV}} = \frac{1}{2}\|\mathbf{u} - \widehat{\mathbf{u}}\|_1 = \frac{1}{2}\sum_{i \in [n]} |u_i - \widehat{u}_i|\,.$$

In the following sections, we will present an algorithm for recovering an estimate $\widehat{\mathbf{w}}$ of $\mathbf{w}$, and give bounds on the error $\|\widehat{\mathbf{w}} - \mathbf{w}\|_{\mathrm{TV}}$ in terms of the problem parameters under natural assumptions on the choice data.

4.3 Accelerated Spectral Ranking Algorithm

In this section, we will describe our algorithm, which we term as accelerated spectral ranking (ASR). Our algorithm is based on the idea of constructing a *random walk*[1] on the *comparison graph* with n vertices, which has an edge between nodes i and j if items i and j are compared in any m-way choice set. The key idea is to construct the random walk such that the probability of transition from node i to node j is proportional to w_j. If w_j is larger than w_i, then with other quantities being equal, one would expect the random walk to spend more time in node j than node i in its steady State distribution. Hence, if we can calculate the stationary distribution of this random walk, it might give us a way to estimate the weight vector $\mathbf{w}$. Moreover, for computational efficiency, we would also want this random walk to have a fast *mixing time*, i.e. it should rapidly converge to its stationary distribution.

The rank centrality (RC) algorithm (Negahban et al., 2017) for the BTL model, and its generalization the Luce spectral ranking (LSR) algorithm (Maystre and Grossglauser, 2015) for the MNL model, are based on a similar idea of constructing a random walk over the comparison graph. These algorithms construct a random walk whose stationary distribution, in expectation, is exactly $\mathbf{w}$. However, this construction forces their Markov chain to have self loops with large mass, slowing down the convergence rate.

In this section we will show that it is possible to design a *significantly* faster mixing random walk that belongs to a different class of random walks over the comparison graph. More precisely, the random walk that we construct is such that it is possible to recover the weight vector $\mathbf{w}$ from its stationary distribution using a fixed linear transformation, while for RC and LSR, the stationary distribution is exactly $\mathbf{w}$. Our theoretical analysis in Section 4.5 as well as experiments on synthetic and real world datasets in Section 4.7 will show that this difference can lead to vastly improved results.

Given choice data $\mathbf{Y}$, let us denote by $G_c([n], E)$ the undirected graph on n vertices, with an

[1] Throughout this chapter we will use the terminology Markov chain and random walk interchangeably.

Algorithm 5 ASR

```
**Input** Markov chain $\widehat{\mathbf{P}}$ according to Eq. (4.3.2)
**Initialize** $\widehat{\boldsymbol{\pi}} = (\frac{1}{n}, \cdots, \frac{1}{n})^\top \in \Delta_n$
**while** estimates do not converge **do**
    $\widehat{\boldsymbol{\pi}} \leftarrow \widehat{\mathbf{P}}^\top \widehat{\boldsymbol{\pi}}$
**end while**
**Output** $\widehat{\mathbf{w}} = \frac{\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}}{\|\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}\|_1}$
```

edge $(i, j) \in E$ for any i, j that are a part of an m-way choice set. More formally, $(i, j) \in E$ if there exists an index $a \in [d]$ such that $i, j \in S_a$. We will call G_c the *comparison* graph, and throughout this chapter, we will assume that $\mathbf{Y}$ is such that G_c is connected. We will denote by d_i the number of unique m-way choice sets of which $i \in [n]$ was a part, i.e. $d_i = \sum_{a \in [d]} \mathbf{1}[i \in S_a]$. Let $\mathcal{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix, with D_{ii} being equal to d_i, $\forall i \in [n]$. Also, let $d_{\max} := \max_i d_i$ and $d_{\min} := \min_i d_i$.

Suppose for each $a \in [d]$ and $j \in S_a$, one had access to the true probability $p_{j|S_a}$ of j being the most preferred item in S_a. Then one could define a random walk on G_c with transition probability from node $i \in [n]$ to $j \in [n]$ given by

$$P_{ij} := \frac{1}{d_i} \sum_{a \in [d]: i,j \in S_a} p_{j|S_a} = \frac{1}{d_i} \sum_{a \in [d]: i,j \in S_a} \frac{w_j}{\sum_{j' \in S_a} w_{j'}} \, . \tag{4.3.1}$$

Let $\mathbf{P} := [P_{ij}]$. One can verify that $\mathbf{P}$ corresponds to a valid transition probability matrix as it is non-negative and row stochastic. Furthermore, $\mathbf{P}$ defines a reversible Markov chain as it satisfies the detailed balance equations

$$w_i \, d_i \, P_{ij} = w_j \, d_j \, P_{ji} \, ,$$

for all $i, j \in [n]$. If the graph G_c is connected then $\boldsymbol{\pi} = \mathcal{D}\,\mathbf{w}/\|\mathcal{D}\,\mathbf{w}\|_1$ is the unique stationary distribution of $\mathbf{P}$, and one can recover the true weight vector $\mathbf{w}$ from this stationary distribution using a linear transform $\mathcal{D}^{-1}$.

In practice one does not have access to $\mathbf{P}$, so we propose an *empirical* estimate of $\mathbf{P}$ that can be computed from the given choice data. Formally, define $\widehat{p}_{i|S_a}$ to be the fraction of times that i was chosen amongst items in the set S_a, i.e. $\widehat{p}_{i|S_a} := \frac{1}{L}\sum_{l=1}^{L} \mathbf{1}[y_a^l = i]$. Let us then define a random walk where the probability of transition from node $i \in [n]$ to node $j \in [n]$ is given by

$$\widehat{P}_{ij} := \frac{1}{d_i} \sum_{a \in [d]: i,j \in S_a} \widehat{p}_{j|S_a} \,. \tag{4.3.2}$$

Let $\widehat{\mathbf{P}} := [\widehat{P}_{ij}]$. One can again verify that $\widehat{\mathbf{P}}$ corresponds to a valid transition probability matrix. We can think of $\widehat{\mathbf{P}}$ as a perturbation of $\mathbf{P}$, with the error due to perturbation decreasing with more and more choices. There is a rich literature (Cho and Meyer, 2001; Mitrophanov, 2005) on analyzing sensitivity of the stationary distribution of a Markov chain under small perturbations. Hence, given a large number of choices, one can expect the stationary distribution of $\widehat{\mathbf{P}}$ to be close to that of $\mathbf{P}$. Since we take a linear transform of these stationary distributions, one also needs to show that closeness is preserved under this linear transform. We defer this analysis to Section 4.5.

The pseudo-code for our algorithm is given in Algorithm 5. The algorithm computes the stationary distribution $\widehat{\boldsymbol{\pi}}$ of the Markov chain $\widehat{\mathbf{P}}$ using the power method.[2] It then outputs the (normalized) vector $\widehat{\mathbf{w}}$ that is obtained after applying the linear transform $\mathcal{D}^{-1}$ to $\widehat{\boldsymbol{\pi}}$, i.e. $\widehat{\mathbf{w}} = \frac{\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}}{\|\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}\|_1}$. In the next section we will compare the convergence time of our algorithm with previous algorithms (Negahban et al., 2017; Maystre and Grossglauser, 2015).

[2] The stationary distribution of the Markov chain may also be computed using other linear algebraic techniques, but these techniques typically have a running time of $O(n^3)$ which is impractical for most modern applications.

4.4 Comparison of Mixing Time with Rank Centrality (RC) and Luce Spectral Ranking (LSR)

The random walk $\mathbf{P}^{\mathrm{RC}}$ constructed by the RC (Negahban et al., 2017) algorithm for the BTL model is given by

$$P_{ij}^{\mathrm{RC}} := \begin{cases} \frac{1}{d_{\max}} \sum_{a \in [d]: i,j \in S_a} p_{j|S_a} & \text{if } i \neq j \\ 1 - \frac{1}{d_{\max}} \sum_{j' \neq i} P_{ij'}^{\mathrm{RC}} & \text{if } i = j \end{cases}, \tag{4.4.1}$$

and the random walk $\mathbf{P}^{\mathrm{LSR}}$ constructed by LSR (Maystre and Grossglauser, 2015) for the MNL model is given by

$$P_{ij}^{\mathrm{LSR}} := \begin{cases} \epsilon \sum_{a \in [d]: i,j \in S_a} p_{j|S_a} & \text{if } i \neq j \\ 1 - \epsilon \sum_{j' \neq i} P_{ij'}^{\mathrm{LSR}} & \text{if } i = j \end{cases}, \tag{4.4.2}$$

where $\epsilon > 0$ is chosen such that the diagonal entries are non-negative. In general ϵ would be $O(\frac{1}{d_{\max}})$. The random walks $\widehat{\mathbf{P}}^{\mathrm{RC}}$ and $\widehat{\mathbf{P}}^{\mathrm{LSR}}$ constructed from the choice data are defined analogously using empirical probabilities $\widehat{p}_{j|S_a}$ instead of $p_{j|S_a}$.

We first begin by showing that for any given choice data $\mathbf{Y}$, both RC/LSR and our algorithm will return the same estimate upon convergence.

Proposition 4.4.1. *Given items $[n]$ and choice data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^{d}$, let $\widehat{\boldsymbol{\pi}}$ be the stationary distribution of the Markov chain $\widehat{\mathbf{P}}$ constructed by* ASR*, and let $\widehat{\mathbf{w}}^{LSR}$ be the stationary distribution of the Markov chain $\widehat{\mathbf{P}}^{LSR}$. Then $\widehat{\mathbf{w}}^{LSR} = \frac{\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}}{\|\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}\|_1}$. The same result is also true for $\widehat{\mathbf{w}}^{RC}$ for the case of pairwise choices.*

Proof. Consider the estimates $\widehat{\mathbf{w}} = \mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}/\|\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}\|_1$ returned by the ASR algorithm upon convergence. In order to prove this lemma it is sufficient to prove that $\mathcal{D}\widehat{\mathbf{w}}^{\mathrm{LSR}}$ is an invariant measure (an eigenvector associated with eigenvalue 1) of the Markov chain $\widehat{\mathbf{P}}$ corresponding to the ASR algorithm.

Since $\widehat{\mathbf{w}}^{\mathrm{LSR}}$ is the stationary distribution (also an eigenvector corresponding to eigenvalue 1) of $\widehat{\mathbf{P}}^{\mathrm{LSR}}$, we have

$$\widehat{\mathbf{w}}^{\mathrm{LSR}} = (\widehat{\mathbf{P}}^{\mathrm{LSR}})^\top \widehat{\mathbf{w}}^{\mathrm{LSR}}.$$

Following the definition (Eq. (4.4.2)) of $\widehat{\mathbf{P}}^{\mathrm{LSR}}$, we have the following relation for all $1 \le i \le n$

$$\begin{aligned} \widehat{w}_i^{\mathrm{LSR}} = \widehat{w}_i^{\mathrm{LSR}} &\left(1 - \epsilon \sum_{j \neq i} \sum_{a: i,j \in S_a} p_{j|S_a}\right) \\ &+ \epsilon \sum_{j \neq i} \sum_{a: i,j \in S_a} p_{j|S_a} \widehat{w}_j^{\mathrm{LSR}} \\ \implies &\sum_{j \neq i} \sum_{a: i,j \in S_a} p_{j|S_a} \widehat{w}_i^{\mathrm{LSR}} = \sum_{j \neq i} \sum_{a: i,j \in S_a} p_{j|S_a} \widehat{w}_j^{\mathrm{LSR}}\,. \end{aligned}$$

We shall use this relation to prove that $\widehat{\mathbf{P}}^\top \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}} = \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}$, where $\widehat{\mathbf{P}}$ is the transition matrix corresponding to the Markov chain constructed by ASR. Consider the i^{th} coordinate $[\widehat{\mathbf{P}}^\top \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}]_i$ of the vector $\widehat{\mathbf{P}}^\top \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}$

$$\begin{aligned} [\widehat{\mathbf{P}}^\top \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}]_i &= \frac{1}{d_i} \sum_{a: i \in S_a} p_{i|S_a} d_i \widehat{w}_i^{\mathrm{LSR}} \\ &\quad + \sum_{j \neq i} \frac{1}{d_j} \sum_{b: i,j \in S_b} p_{j|S_b} d_j \widehat{w}_j^{\mathrm{LSR}} \\ &= \sum_{a: i \in S_a} p_{i|S_a} \widehat{w}_i^{\mathrm{LSR}} + \sum_{j \neq i} \sum_{b: i,j \in S_b} p_{j|S_b} \widehat{w}_i^{\mathrm{LSR}} \\ &= \sum_{a: i \in S_a} (\sum_{j \in S_a} p_{j|S_a}) \widehat{w}_i^{\mathrm{LSR}} \\ &= \sum_{a: i \in S_a} \widehat{w}_i^{\mathrm{LSR}} \\ &= d_i \widehat{w}_i^{\mathrm{LSR}} = [\mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}]_i\,, \end{aligned}$$

where the second equality follows from the relation we proved earlier. Furthermore, this identity holds for all $1 \le i \le n$, from which we can conclude $\widehat{\mathbf{P}}^\top \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}} = \mathcal{D} \widehat{\mathbf{w}}^{\mathrm{LSR}}$. Furthermore, if the respective Markov chains induced by the choice data are ergodic, then the corresponding stationary distributions must be unique, which is sufficient to prove both LSR and ASR return the same estimates upon convergence.

Since Luce spectral ranking is a generalization of the rank centrality algorithm, the transition matrix $\widehat{\mathbf{P}}^{\mathrm{LSR}}$ is identical to the transition matrix $\widehat{\mathbf{P}}^{\mathrm{RC}}$ in the pairwise choice setting after setting $\epsilon = \frac{1}{d_{\max}}$, and thus, we can also conclude $\widehat{\mathbf{P}}^{\top}\mathcal{D}\widehat{\mathbf{w}}^{\mathrm{RC}} = \mathcal{D}\widehat{\mathbf{w}}^{\mathrm{RC}}$. Thus, the statement of the lemma follows. □

Although the above lemma shows that in a convergent state both these algorithms will return the same estimates, it does not say anything about the time it takes to reach this convergent State. This is where the *key difference* lies.

Observe that each row $i \in [n]$ of our matrix $\mathbf{P}$ is divided by d_i, whereas each row of $\mathbf{P}^{\mathrm{RC}}$ is divided by $d_{\max}$ except the diagonal entries. Now if $d_{\max}$ is very large, a row $i \in [n]$ of $\mathbf{P}^{\mathrm{RC}}$ that corresponds to an item i with small d_i would have very small non-diagonal entries. This can make the diagonal entry P_{ii}^{RC} very large, which amounts to having a heavy self loop at node i. This heavy self loop can significantly reduce the time it takes for the random walk to reach its stationary distribution, since a lot of transitions starting from i will return back to i. The same analysis holds true for LSR under multiway choices.

To formalize this intuition, we need to analyze the spectral gap of a random walk $\mathcal{X}$, which we denote by $\mu(\mathcal{X})$, which plays an important role in determining its mixing time. The spectral gap of a reversible random walk (or Markov chain) $\mathcal{X}$ is defined as $\mu(\mathcal{X}) := 1 - \lambda_2(\mathcal{X})$, where $\lambda_2(\mathcal{X})$ is the second largest eigenvalue of $\mathcal{X}$ in terms of absolute value. The following lemma (see Levin et al. (2008) for more details) gives both upper and lower bounds on the mixing time (w.r.t. the total variation distance) of a random walk in terms of the spectral gap.

Lemma 4.4.2. *(Levin et al., 2008) Let $\mathbf{X}$ be the transition probability matrix of a reversible, irreducible Markov chain with State space $[n]$, $\boldsymbol{\pi}$ be the stationary distribution of $\mathbf{X}$, and*

$\pi_{\min} := \min_{i\in[n]} \pi_i$, and let

$$d(r) = \sup_{\mathbf{p}\in\Delta_n} \|\mathbf{p}\mathbf{X}^r - \boldsymbol{\pi}\|_{TV}.$$

For any $\gamma > 0$, let $t(\gamma) = \min\{r \in \mathbb{N} : d(r) \leq \gamma\}$; then

$$\log(\frac{1}{2\gamma})\Big(\frac{1}{\mu(\mathbf{X})} - 1\Big) \leq t(\gamma) \leq \log(\frac{1}{\gamma\pi_{\min}})\frac{1}{\mu(\mathbf{X})}.$$

The above lemma States that the mixing time of a Markov chain $\mathbf{X}$ is inversely proportional to its spectral gap $\mu(\mathbf{X})$. Now, we will compare the spectral gap of our Markov chain $\mathbf{P}$ with the spectral gap of $\mathbf{P}^{\text{RC}}$ (and $\mathbf{P}^{\text{LSR}}$).

Proposition 4.4.3. *Let the probability transition matrix $\mathbf{P}$ for our random walk be as defined in Eq. (4.3.1). Let $\mathbf{P}^{RC}$ and $\mathbf{P}^{LSR}$ be as defined in Eq. (4.4.1) and Eq. (4.4.2), respectively. Then*

$$\frac{d_{\min}}{d_{\max}}\mu(\mathbf{P}) \leq \mu(\mathbf{P}^{RC}) \leq \mu(\mathbf{P}), \tag{4.4.3}$$

and

$$\epsilon d_{\min}\mu(\mathbf{P}) \leq \mu(\mathbf{P}^{LSR}) \leq \mu(\mathbf{P}), \tag{4.4.4}$$

where $\epsilon = O(\frac{1}{d_{\max}})$.

Lemma 4.4.4. *(Diaconis and Saloff-Coste, 1993) Let $\mathbf{Q}$ and $\mathbf{P}$ be reversible Markov chains on a finite set $[n]$ representing random walks on a graph $G = ([n], E)$, i.e. $P_{ij} = Q_{ij} = 0$ for all $(i,j) \notin E$. Let $\boldsymbol{\nu}$ and $\boldsymbol{\pi}$ be the stationary distributions of $\mathbf{Q}$ and $\mathbf{P}$, respectively. Then the spectral gaps of $\mathbf{Q}$ and $\mathbf{P}$ are related as*

$$\frac{\mu(\mathbf{P})}{\mu(\mathbf{Q})} \geq \frac{\alpha}{\beta}$$

where $\alpha := \min_{(i,j)\in E}\{\pi_i P_{ij}/\nu_i Q_{ij}\}$ and $\beta := \max_{i\in[n]}\{\pi_i/\nu_i\}$.

We are now ready to prove Proposition 4.4.3.

Proof. (of Proposition 4.4.3) To prove this lemma, we shall leverage the above comparison lemma due to Diaconis and Saloff-Coste (1993), that compares the spectral gaps of two arbitrary reversible Markov Chains. Let $\mathbf{P}$ (Eq. (4.3.2)) be the reversible Markov chain corresponding to ASR with stationary distribution $\boldsymbol{\pi} = \mathcal{D}\mathbf{w}/\|\mathcal{D}\mathbf{w}\|_1$, and let $\mathbf{P}^{\text{LSR}}$ (Eq. (4.4.2)) be the reversible Markov chain corresponding to LSR (RC in the pairwise case) with stationary distribution $\boldsymbol{\pi}^{\text{LSR}}$. Then by Lemma 4.4.4,

$$\frac{\mu(\mathbf{P}^{\text{LSR}})}{\mu(\mathbf{P})} \geq \frac{\alpha}{\beta}$$

where

$$\begin{aligned} \alpha &:= \min_{(i,j):\exists a \text{ s.t. } i,j\in S_a} \left(\frac{\pi_i^{\text{LSR}} P_{ij}^{\text{LSR}}}{\pi_i P_{ij}} \right), \\ \beta &:= \max_{i\in[n]} \left(\frac{\pi_i^{\text{LSR}}}{\pi_i} \right). \end{aligned}$$

From the definition of $\mathbf{P}$, and $\mathbf{P}^{\text{LSR}}$, we have

$$\begin{aligned} \mathbf{P}_{ij} &= \frac{1}{d_i} \sum_{a\in[d]:i,j\in S_a} \frac{w_j}{\sum_{k\in S_a} w_k}, \\ \mathbf{P}_{ij}^{\text{LSR}} &= \epsilon \sum_{a\in[d]:i,j\in S_a} \frac{w_j}{\sum_{k\in S_a} w_k} \end{aligned}$$

From the above equations and Proposition 4.4.1, it is easy to see that

$$\begin{aligned} \alpha &= \epsilon\|\mathcal{D}\mathbf{w}\|_1, \quad \text{and} \\ \beta &= \frac{\|\mathcal{D}\mathbf{w}\|_1}{d_{\min}} \\ \implies \mu(\mathbf{P}^{\text{LSR}}) &\geq \epsilon d_{\min}(\mu(\mathbf{P})) \end{aligned}$$

Following an identical line of reasoning, we have

$$\frac{\mu(\mathbf{P})}{\mu(\mathbf{P}^{\text{LSR}})} \geq \frac{\alpha'}{\beta'}$$

where

$$\alpha' = \min_{(i,j):\exists a \text{ s.t. } i,j \in S_a} \left(\frac{\pi_i P_{ij}}{\pi_i^{\text{LSR}} P_{ij}^{\text{LSR}}} \right),$$
$$\beta' = \max_{i \in [n]} \left(\frac{\pi_i}{\pi_i^{\text{LSR}}} \right)$$

From the definition of $\mathbf{P}$, and $\mathbf{P}^{\text{LSR}}$, we have

$$\begin{aligned} \alpha' &= \frac{1}{\|\mathcal{D}\mathbf{w}\|_1 \epsilon}, \quad \text{and} \\ \beta' &= \frac{d_{\max}}{\|\mathcal{D}\mathbf{w}\|_1} \\ \implies \mu(\mathbf{P}) &\geq \frac{1}{\epsilon d_{\max}} (\mu(\mathbf{P}^{\text{LSR}}))\,. \end{aligned}$$

Since $\epsilon \leq 1/d_{\max}$, we get the following comparison between the spectral gaps of the Markov chains corresponding to the two approaches

$$\epsilon d_{\min} \mu(\mathbf{P}) \leq \mu(\mathbf{P}^{\text{LSR}}) \leq \mu(\mathbf{P})\,.$$

The same analysis works for the Markov chain $\mathbf{P}^{\text{RC}}$ constructed by rank centrality for the pairwise comparison case with $\epsilon = 1/d_{\max}$, from which we can conclude

$$\frac{d_{\min}}{d_{\max}} \mu(\mathbf{P}) \leq \mu(\mathbf{P}^{\text{RC}}) \leq \mu(\mathbf{P})\,.$$

□

This lemma shows that the spectral gap of $\mathbf{P}$ is always lower bounded by that of $\mathbf{P}^{\text{RC}}$ (and $\mathbf{P}^{\text{LSR}}$), but can be much larger than it. In the latter case one would observe, using Lemma 4.4.2, that our algorithm will converge faster than the RC algorithm (and LSR). In fact there are instances where $O(d_{\max}/d_{\min}) = \Omega(n)$ and the leftmost inequalities in both Eq. (4.4.3) and Eq. (4.4.4) hold with equality. In these instances the convergence of our algorithm will be $\Omega(n)$ times faster. We give examples of two such instances.

Example 4.4.5. *Let $n = 3$, $m = 2$, $w_1 = 1/2, w_2 = 1/4$ and $w_3 = 1/4$. In the choice data* 1 *is compared to both* 2 *and* 3*; but items* 2 *and* 3 *are not compared to each other. This implies that $d_1 = 2$, and $d_i = 1$ for $i \neq 1$. One can calculate the matrices $\mathbf{P}$ and $\mathbf{P}^{RC}$, and their respective eigenvalues, and observe that $\mu(\mathbf{P}) = 2\mu(\mathbf{P}^{RC})$.*

Example 4.4.6. *Let $m = 2$, $\mathbf{w} = (1/n, \cdots, 1/n)^\top$, and the choice data be such that item* 1 *is compared to every other item, and no other items are compared to each other. This implies that $d_1 = n - 1$, and $d_i = 1$ for $i \neq 1$. One can calculate the matrix $\mathbf{P}$ and $\mathbf{P}^{RC}$ again, and their respective eigenvalues, and observe that $\mu(\mathbf{P}) = (n-1) \cdot \mu(\mathbf{P}^{RC})$.*

Note that in the above lemma, we only show the relation between the spectral gaps of the matrices $\mathbf{P}$ and $\mathbf{P}^{\mathrm{RC}}$, and not for any particular realization $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{P}}^{\mathrm{RC}}$. If the Markov chains $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{P}}^{\mathrm{RC}}$ are reversible, then identical results hold. However, similar results are very hard to prove for non-reversible Markov chains (Dyer et al., 2006). Nevertheless, for large L, one can expect the realized matrices $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{P}}^{\mathrm{RC}}$ to be close to their expected matrices $\mathbf{P}$ and $\mathbf{P}^{\mathrm{RC}}$, respectively. Hence, using eigenvalue perturbation bounds (Horn and Johnson, 1990), one can show that the spectrum of $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{P}}^{\mathrm{RC}}$ is close to the spectrum of $\mathbf{P}$ and $\mathbf{P}^{\mathrm{RC}}$, respectively. The same analysis holds true for LSR under multiway choices. In Section 4.7 we perform experiments on synthetic and real world datasets which empirically show that the mixing times of the realized Markov chains behave as predicted.

It has been observed that faster mixing rates of Markov chains gives us the ability to prove sharper perturbation bounds for these Markov chains (Mitrophanov, 2005). In the following section we will use these perturbation bounds to prove sharper sample complexity bounds for our algorithm.

4.5 Sample Complexity Bounds

In this section we will present sample complexity bounds for the estimates returned by ASR in terms of total variation distance. The following theorem gives an error bound in terms of the total variation distance for estimates $\widehat{\mathbf{w}}$ of the MNL weights returned by our algorithm

Theorem 4.5.1. *Given items $[n]$ and choice data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^d$, let each set S_a of cardinality m be compared L times, with outcomes $\mathbf{y}_a = (y_a^1, \cdots, y_a^L)$ produced as per a MNL model with parameters $\mathbf{w} = (w_1, \ldots, w_n)$, such that $\|\mathbf{w}\|_1 = 1$. If the random walk $\widehat{\mathbf{P}}$ (Eq. (4.3.2)) on the comparison graph $G_c([n], E)$ induced by the choice data $\mathbf{Y}$ is strongly connected, then the* ASR *algorithm (Algorithm 5) converges to a unique distribution $\widehat{\mathbf{w}}$, which with probability $\geq 1 - 3n^{-(C^2-50)/25}$ satisfies the following error bound*[3]

$$\|\mathbf{w} - \widehat{\mathbf{w}}\|_{TV} \leq \frac{C\,\kappa\, d_{\text{avg}}}{\mu(\mathbf{P})\, d_{\text{min}}} \sqrt{\frac{\max\{m, \log(n)\}}{L}}\,,$$

where $\kappa = \log\left(\frac{d_{\text{avg}}}{d_{\text{min}} w_{\text{min}}}\right)$, $w_{\text{min}} = \min_{i\in[n]} w_i$, $d_{\text{avg}} = \sum_{i\in[n]} w_i d_i$, $d_{\text{min}} = \min_{i\in[n]} d_i$, $\mu(\mathbf{P})$ is the spectral gap of the random walk $\mathbf{P}$ (Eq. (4.3.1)), and C is any constant.

Let us start by stating some auxiliary lemmas that are needed for the proof of the above theorem.

Lemma 4.5.2 (Multinomial distribution inequality)**.** *(Devroye, 1983) Let $Y_1, \ldots, Y_n$ be a sequence of n independent random variables drawn from the multinomial distribution with parameters $(p_1, \ldots, p_k)$. Let X_i be the number of times i occurs in the n draws, i.e. $X_i = \sum_{j=1}^n \mathbf{1}[Y_j = i]$. For all $\epsilon \in (0, 1)$, and all k satisfying $k/n \leq \epsilon^2/20$, we have*

$$P(\sum_{i=1}^k |X_i - np_i| \geq n\epsilon) \leq 3\exp(-n\epsilon^2/25).$$

To prove Theorem 4.5.1, we shall first prove a bound on the total variation distance between the stationary states $\boldsymbol{\pi}$ and $\widehat{\boldsymbol{\pi}}$ of the transition matrices $\mathbf{P}$ and $\widehat{\mathbf{P}}$ respectively. We shall then prove a bound on the distance between the true weights $\mathbf{w}$ and estimates $\widehat{\mathbf{w}}$ in terms of the distance between $\boldsymbol{\pi}$ and $\widehat{\boldsymbol{\pi}}$.

[3] The dependence on κ is due to the dependence on $\frac{1}{\pi_{\text{min}}}$ in the mixing time upper bounds in Lemma 4.4.2. There are other bounds for κ in terms of the condition number for Markov chains, for example see (Mitrophanov, 2005), and any improvement on these bounds will lead to an improvement in our sample complexity. In the worst case, κ has a trivial upper bound of $O(\log n)$.

An important result in the stability theory of Markov chains shows a connection between the stability of a chain and its speed of convergence to equilibrium (Mitrophanov, 2005). In fact, we can bound the sensitivity of a Markov chain under perturbation as a function of the convergence rate of the chain, with the accuracy of the sensitivity bound depending on the sharpness of the bound on the convergence rate. The following theorem is a specialization of Theorem 3.1 of Mitrophanov (2005), which gives perturbation bounds for Markov chains with general state spaces.

Theorem 4.5.3. *(Mitrophanov, 2005) Consider two discrete-time Markov chains* $\mathbf{P}$ *and* $\widehat{\mathbf{P}}$*, with finite state space* $\Omega = \{1, \ldots, n\}, n \geq 1$*, and stationary distributions* $\boldsymbol{\pi}$ *and* $\widehat{\boldsymbol{\pi}}$*, respectively. If there exist positive constants* $1 < R < \infty$ *and* $\rho < 1$ *such that*

$$\max_{x \in \Omega} \|\mathbf{P}^t(x, \cdot) - \boldsymbol{\pi}\|_{TV} \leq R\rho^t, \qquad \forall t \in \mathbb{N}$$

then for $\mathbf{E} := \mathbf{P} - \widehat{\mathbf{P}}$*, we have*

$$\|\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}\|_{TV} \leq \left(\widehat{t} + \frac{1}{1-\rho}\right) \cdot \|\mathbf{E}\|_{\infty} .$$

where $\widehat{t} = \log(R)/\log(1/\rho)$*, and* $\| \cdot \|_{\infty}$ *is the matrix norm induced by the* L_{∞} *vector norm.*

It is well known that all ergodic Markov chains satisfy the conditions imposed by Theorem 4.5.3. In order to obtain sharp bounds on the convergence rate, we shall leverage the fact that the (unperturbed) Markov chain corresponding to the ideal transition probability matrix $\mathbf{P}$ is time-reversible.

Theorem 4.5.4. *(Diaconis and Stroock, 1991) Let* $\mathbf{P}$ *be an irreducible, reversible Markov chain with finite state space* $\Omega = \{1, \ldots, n\}, n \geq 1$*, and stationary distribution* $\boldsymbol{\pi}$*. Let* $\lambda_2 := \lambda_2(\mathbf{P})$ *be the second largest eigenvalue of* $\mathbf{P}$ *in terms of absolute value. Then for all* $x \in \Omega$*,* $t \in \mathbb{N}$*,*

$$\|\mathbf{P}^t(x, \cdot) - \boldsymbol{\pi}\|_{TV} \leq \sqrt{\frac{1-\pi(x)}{4\pi(x)}}\lambda_2^t$$

Comparing these bounds with the conditions imposed by Theorem 4.5.3, we can observe that

$$
\begin{aligned}
\rho &= \lambda_2, \\
R &= \max_{i\in[n]} \sqrt{\frac{1-\pi(i)}{4\pi(i)}} \\
&= \max_{i\in[n]} \sqrt{\frac{\|\mathcal{D}\mathbf{w}\|_1 - w_i d_i}{4 w_i d_i}} \\
&\leq \sqrt{\frac{d_{\text{avg}}}{4 d_{\min} w_{\min}}},
\end{aligned}
$$

where $w_{\min} = \min_{i\in[n]} w_i$. Substituting these values into the perturbation bounds of Theorem 4.5.3, we get

$$
\begin{aligned}
\widehat{t} + \frac{1}{1-\rho} &= \frac{\log(d_{\text{avg}}/(4 d_{\min} w_{\min}))}{2\log(1/\lambda_2(\mathbf{P}))} + \frac{1}{1-\lambda_2(\mathbf{P})} \\
&\leq \frac{\log(d_{\text{avg}}/(4 d_{\min} w_{\min}))}{2(1-\lambda_2(\mathbf{P}))} + \frac{1}{1-\lambda_2(\mathbf{P})} \\
&< \frac{\kappa}{2\mu(\mathbf{P})}, \qquad \text{where } \kappa = \log(\frac{2 d_{\text{avg}}}{d_{\min} w_{\min}})
\end{aligned}
$$

Now, the next step is to show that the perturbation error $\mathbf{E} := \mathbf{P} - \widehat{\mathbf{P}}$ is bounded in terms of the matrix L_∞ norm.

Lemma 4.5.5. *For* $\mathbf{E} := \mathbf{P} - \widehat{\mathbf{P}}$, *we have with probability* $\geq 1 - 3n^{-(C^2-50)/25}$,

$$
\|\mathbf{E}\|_\infty \leq C\sqrt{\frac{\max\{m, \log n\}}{L}}
$$

where C *is any constant.*

Proof. By definition, $\|\mathbf{E}\|_\infty = \max_i \sum_{j=1}^n |\widehat{P}_{ij} - P_{ij}|$. Fix any row $i \in [n]$. The probability that the absolute row sum exceeds a fixed positive quantity t is given by

$$\begin{aligned}
&P(\sum_{j=1}^{n} |\widehat{P}_{ij} - P_{ij}| \geq t) \\
&= P(\sum_{j=1}^{n} |\frac{1}{d_i} \sum_{a:i,j\in S_a} (\widehat{p}_{j|S_a} - p_{j|S_a})| \geq t) \\
&= P(\sum_{j=1}^{n} |\frac{1}{d_i} \sum_{a:i,j\in S_a} \frac{1}{L} \sum_{l=1}^{L} (\mathbf{1}(y_a^l = j) - p_{j|S_a})| \geq t) \\
&\leq P(\sum_{j=1}^{n} \sum_{a:i,j\in S_a} |\sum_{l=1}^{L} (\mathbf{1}(y_a^l = j) - p_{j|S_a})| \geq L d_i t) \\
&= P(\sum_{a:i\in S_a} \sum_{j\in S_a} |\sum_{l=1}^{L} (\mathbf{1}(y_a^l = j) - p_{j|S_a})| \geq L d_i t) \\
&\leq d_i P(\sum_{j\in S_a} |\sum_{l=1}^{L} (\mathbf{1}(y_l^a = j) - p_{j|S_a})| \geq \frac{L d_i t}{d_i})
\end{aligned}$$

with the final pair of inequalities following from rearranging the terms in the summations and applying union bound. We leverage the multinomial distribution concentration inequality (Lemma 4.5.2) of Devroye (1983) to obtain the following bound for any set S_a for any m satisfying a technical condition $m/L \leq t^2/20$.

$$P(\sum_{j\in S_a} |\sum_{l=1}^{L} (\mathbf{1}(y_l^a = j) - p_{j|S_a})| \geq Lt) \leq 3\exp(\frac{-Lt^2}{25})$$

Thus, using union bound, the probability that any absolute row sum exceeds t is at most $3nd_{\max}\exp(-Lt^2/25)$. By selection of $t = 5C'\sqrt{\max\{m, \log n\}/L}$, we get

$$\begin{aligned}
&P\left(\|\mathbf{E}\|_\infty \geq 5C'\sqrt{\frac{\max\{m, \log n\}}{L}}\right) \\
&\leq 3n^2 \exp\left(\frac{-25C'^2 L \max\{m, \log n\}}{25L}\right) \\
&\leq 3n^{-(C'^2-2)}
\end{aligned}$$

substituting $C = 5C'$ proves our claim. Lastly, one can verify that the aforementioned choice of t satisfies the technical condition imposed by Lemma 4.5.2 for any n, m and L. $\square$

Combining the results of Theorem 4.5.3, Theorem 4.5.4, and Theorem 4.5.5 gives us a high confidence total variation error bound on the stationary states $\boldsymbol{\pi}$ and $\widehat{\boldsymbol{\pi}}$ of the ideal and perturbed Markov chains $\mathbf{P}$ and $\widehat{\mathbf{P}}$ respectively. Thus, with confidence $\geq 1 - 3n^{-(C^2-50)/25}$, we have

$$\|\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}\|_{\mathrm{TV}} \leq \frac{C\kappa}{\mu(\mathbf{P})}\sqrt{\frac{\max\{m, \log n\}}{L}}\,, \tag{4.5.1}$$

where $\kappa = \log(2d_{\text{avg}}/(d_{\min}w_{\min}))$.

The last step in our scheme is to prove that the linear transformation $\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}$ preserves this error bound up to a reasonable factor.

Lemma 4.5.6. *Under the conditions of Theorem 4.5.1, let* $\boldsymbol{\pi} = \mathcal{D}\mathbf{w}/\|\mathcal{D}\mathbf{w}\|_1$ *and* $\widehat{\boldsymbol{\pi}} = \mathcal{D}\widehat{\mathbf{w}}/\|\mathcal{D}\widehat{\mathbf{w}}\|_1$ *be the unique stationary distributions of the Markov chains* $\mathbf{P}$ *(Eq. (4.3.1)) and* $\widehat{\mathbf{P}}$ *(Eq. (4.3.2)) respectively. Then we have*

$$\|\mathbf{w} - \widehat{\mathbf{w}}\|_{TV} \leq \frac{d_{\text{avg}}}{d_{\min}}\|\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}\|_{TV}.$$

Proof. We shall divide our proof into two cases.

Case 1: $\|\mathcal{D}\widehat{\mathbf{w}}\|_1 \geq \|\mathcal{D}\mathbf{w}\|_1$.

Let us define the set $A = \{i : w_i \geq \widehat{w}_i\}$, and the set $A' = \{j : \pi_j \geq \widehat{\pi}_j\}$. When $\|\mathcal{D}\widehat{\mathbf{w}}\|_1 \geq \|\mathcal{D}\mathbf{w}\|_1$, it is easy to see that $A \subseteq A'$.

Consider the total variation distance $\|\mathbf{w} - \widehat{\mathbf{w}}\|_{TV}$ between the true preferences $\mathbf{w}$ and our estimates $\widehat{\mathbf{w}}$. By definition,

$$\begin{aligned}
\|\mathbf{w}-\widehat{\mathbf{w}}\|_{TV} &= \sum_{i\in A}(w_i-\widehat{w}_i) \\
&= \sum_{i\in A} w_i\left(1-\frac{\widehat{w}_i}{w_i}\right) = \sum_{i\in A} w_i\left(1-\frac{\widehat{w}_i d_i}{w_i d_i}\right) \\
&\leq \sum_{i\in A} w_i\left(1-\frac{\widehat{w}_i d_i\|\mathcal{D}\mathbf{w}\|_1}{w_i d_i\|\mathcal{D}\widehat{\mathbf{w}}\|_1}\right) \\
&= \sum_{i\in A} w_i\left(1-\frac{\widehat{\pi}_i}{\pi_i}\right) \\
&= \sum_{i\in A} w_i\left(\frac{(\pi_i-\widehat{\pi}_i)\|\mathcal{D}\mathbf{w}\|_1}{w_i d_i}\right) \\
&\leq \sum_{j\in A'} w_j\left(\frac{(\pi_j-\widehat{\pi}_j)\|\mathcal{D}\mathbf{w}\|_1}{w_j d_j}\right) \\
&= \sum_{j\in A'}\left(\frac{(\pi_j-\widehat{\pi}_j)\|\mathcal{D}\mathbf{w}\|_1}{d_j}\right) \\
&\leq \frac{\|\mathcal{D}\mathbf{w}\|_1}{d_{\text{min}}}\sum_{j\in A'}(\pi_j-\widehat{\pi}_j) = \frac{d_{\text{avg}}}{d_{\text{min}}}\|\pi-\widehat{\pi}\|_{TV}
\end{aligned}$$

Case 2, where $\|\mathcal{D}\widehat{\mathbf{w}}\|_1 < \|\mathcal{D}\mathbf{w}\|_1$ follows symmetrically, giving us the inequality

$$\begin{aligned}
\|\mathbf{w}-\widehat{\mathbf{w}}\|_{TV} &\leq \frac{\|\mathcal{D}\widehat{\mathbf{w}}\|_1}{d_{\text{min}}}\|\pi-\widehat{\pi}\|_{TV} \\
&\leq \frac{\|\mathcal{D}\mathbf{w}\|_1}{d_{\text{min}}}\|\pi-\widehat{\pi}\|_{TV} = \frac{d_{\text{avg}}}{d_{\text{min}}}\|\pi-\widehat{\pi}\|_{TV}
\end{aligned}$$

where the last inequality follows from the assumption of Case 2, proving our claim. □

Proof. (of Theorem 4.5.1) The theorem follows easily by combining the above lemma with Eq. (4.5.1). □

In the error bound of Theorem 4.5.1, one can further bound the spectral gap $\mu(\mathbf{P})$ of $\mathbf{P}$ in terms of the spectral gap of the *random walk normalized Laplacian* of G_c, which is a fundamental quantity associated with G_c. The Laplacian represents a random walk on G_c

that transitions from a node i to one of its neighbors uniformly at random. Formally, the Laplacian $\mathbf{L} := \mathbf{C}^{-1}\mathbf{A}$, where $\mathbf{C}$ is a diagonal matrix with $C_{ii} = \big|\bigcup_{a\in[d]:i\in S_a} S_a\big|$, i.e. the number of unique items i was compared with, and $\mathbf{A}$ is the adjacency matrix, such that for $i,j \in [n]$, $A_{ij} = 1$ if $(i,j)\in E$, and $A_{ij} = 0$ otherwise. Let $\xi := \mu(\mathbf{L})$ be the spectral gap of $\mathbf{L}$. Then we can lower bound $\mu(\mathbf{P})$ as follows (proof in the Appendix)

$$\mu(\mathbf{P}) \geq \frac{\xi}{m\, b^2}\,,$$

where b is the ratio of the maximum to the minimum weight, i.e. $b = \max_{i,j\in[n]} w_i/w_j$. This gives us the following.

Corollary 4.5.7. *In the setting of Theorem 4.5.1, the* ASR *algorithm converges to a unique distribution* $\widehat{\mathbf{w}}$*, which with probability* $\geq 1 - 3n^{-(C^2-50)/25}$ *satisfies the following error bound:*

$$\|\mathbf{w} - \widehat{\mathbf{w}}\|_{TV} \leq \frac{C\, m\, b^2\, \kappa\, d_{\text{avg}}}{\xi\, d_{\min}} \sqrt{\frac{\max\{m, \log(n)\}}{L}}\,,$$

where $b = \max_{i,j\in[n]} \frac{w_i}{w_j}$.

The proof of the above corollary is given in the Appendix. In the discussion that follows, we will assume $b = O(1)$, and hence, $\mu(\mathbf{P}) = \Omega(\xi/m)$. The quantity d_{avg} has an interesting interpretation: it is the weighted average of the number of sets in which each item was shown. It has a trivial upper bound of $d_{\max}$, however, a careful analysis will reveal a better bound of $O(|E|/n)$ where E is the set of edges in the comparison graph G_c. Using this observation we can give the following corollary of the above theorem.

Corollary 4.5.8. *If the conditions of Theorem 4.5.1 are satisfied, and if the number of edges in the comparison graph* G_c *are* $O(n\,\text{poly}(\log n))$*, i.e.* $|E| = O(n\,\text{poly}(\log n))$*, then in order to ensure a total variation error of* $o(1)$*, the required number of choices per set is upper bounded as*

$$L = O\big(\mu(\mathbf{P})^{-2}\,\text{poly}(\log n)\big) = O\big(\xi^{-2}\, m^3\, \text{poly}(\log n)\big)\,.$$

Hence, the sample complexity, i.e. total number of m-way choices needed to estimate $\mathbf{w}$ *with error* $o(1)$*, is given by* $|E| \times L = O\big(\xi^{-2}\, m^3\, n\, \mathrm{poly}(\log n)\big)$*.*

The proof of the this corollary is given in the Appendix. Note that the case when the total number of edges in the comparison graph is $O(n\, \mathrm{poly}(\log n))$ captures the most interesting case in ranking and sorting. Also, in most practical settings the size m of choice sets will be $O(\log n)$. In this case, the above corollary implies a sample complexity bound of $O\big(\xi^{-2}\, n\, \mathrm{poly}(\log n)\big)$, which is sometimes referred to as *quasi-linear* complexity. The following simple example illustrates this sample complexity bound.

Example 4.5.9. *Consider a star comparison graph, discussed in Example 4.4.6, where there is one item* $i \in [n]$ *that is compared to all other* $n-1$ *items, and no other items are compared to each other. Let* $\mathbf{w} = (\frac{1}{n}, \cdots, \frac{1}{n})^\top$*. One can calculate the spectral gap* $\mu(\mathbf{P})$ *to be* 0.5 *exactly. In this case, the sample complexity bound given by our result is* $O(n\, \mathrm{poly}(\log n))$*.*

Discussion/Comparison. For the special case of pairwise choices under the BTL model ($m = 2$), Negahban et al. (2017) give a sample complexity bound of $O\big(\frac{d_{\max}}{d_{\min}}\xi^{-2}\, n\, \mathrm{poly}(\log n)\big)$ for recovering the estimates $\widehat{\mathbf{w}}$ with low (normalized) L_2 error. Using Proposition 4.4.1 one can see that this bound also applies to the estimates returned by our algorithm, and our bound in terms of L_1 applies to rank centrality as well. However, the bounds due to Negahban et al. (2017) have a dependence on the ratio $\frac{d_{\max}}{d_{\min}}$ due to the large spectral gap of their Markov chain as compared to ξ, the spectral gap of the Laplacian. In Section 4.7 we show that for many real world datasets $\frac{d_{\max}}{d_{\min}}$ can be much larger than $\log n$, and hence, their bounds are no longer quasi-linear. A large class of graphs that occur in many real world scenarios and exhibit this behavior are the power-law graphs. Another real world scenario in which $\frac{d_{\max}}{d_{\min}} = \Omega(n)$ arises is choice modeling (Agrawal et al., 2016), where one explicitly models the *'no choice option'* where the user has an option of not selecting any item from the set of items presented to her. In this case the 'no choice option' will be present in each choice set, and the comparison graph will behave like a star graph discussed in Example 4.4.6. In fact for such graphs, the results of (Negahban et al., 2017) give a trivial bound of poly(n)

Algorithm 6 Message Passing

```
Input Graph $G_f = ([n] \cup [d], E_f)$, edge $(i, a) \in E$ has weight $\widehat{p}_{i|S_a}$
Initialize Set $m^{(0)}_{a \to i} \leftarrow m/n$, $\forall a \in [d]$, $\forall i \in S_a$
for $t = 1, 2, \cdots$ until convergence do
    for all $i \in [n]$ do $m^{(t)}_{i \to a} = \frac{1}{d_i} \sum_{a' : i \in S_{a'}} \widehat{p}_{i|S_{a'}} \cdot m^{(t-1)}_{a' \to i}$
    for all $a \in [d]$ do $m^{(t)}_{a \to i} = \sum_{i' \in S_a} m^{(t)}_{i' \to a}$
end for
Set $\widehat{w}_i \leftarrow m^{(t-1)}_{i \to a}$, $\forall i \in [n]$
Output $\widehat{\mathbf{w}} / \|\widehat{\mathbf{w}}\|_1$
```

in terms of the L_2 error.

For the general case of multiway choices we are not aware of any other sample complexity bounds. It is also important to note that the dependence on the number of choice sets comes only through the spectral gap ξ of the natural random walk on the comparison graph. For example, if the graph is a cycle ($d = n$), then the spectral gap is $O(1/n^2)$, whereas if the graph is a clique ($d = O(n^2)$) the spectral gap is $O(1)$.

4.6 Message Passing Interpretation of ASR

In this section, we show our spectral ranking algorithm can be interpreted as a message passing/belief propagation algorithm. This connection can be used to design a decentralized distributed version of our algorithm.

Let us introduce the *factor graph*, which is an important data structure used in message passing algorithms. The factor graph is a bipartite graph $G_f([n] \cup [d], E_f)$ which has two type of nodes– *item nodes* which correspond to the n items, and *set nodes* which correspond to the d sets. More formally, there is an item node i for each item $i \in [n]$, and there is a set node a for each set S_a, $\forall a \in [d]$. There is an edge $(i, a) \in E_f$ between node i and a if and only if $i \in S_a$. There is a weight $\widehat{p}_{i|S_a}$ on the edge (i, a) which corresponds to the fraction of times i won in the set S_a.

We shall now describe the algorithm. In each iteration of this algorithm, the item nodes send a message to their neighboring set nodes, and the set nodes respond to these messages. A message from an item node i to a set node a represents an estimate of the weight w_i of item i, and a message from a set node a to an item i represents an estimate of the sum of weights of items contained in set S_a.

In each iteration, the item nodes update their estimates based on the messages they receive in the previous iteration, and send these estimates to their neighboring set nodes. The set nodes then update their estimate by summing up the messages they receive from their neighboring item nodes, and then send these estimates to their neighboring item nodes. This process continues until the messages converge.

Formally, let $m_{i \to a}^{(t-1)}$ be the message from item node i to set node a in iteration $t-1$, and $m_{a \to i}^{(t-1)}$ be the corresponding message from the set node a to item node i. Then the messages in the next iteration are updated as follows:

$$m_{i \to a}^{(t)} = \frac{1}{d_i} \sum_{a' \in [d] : i \in S_{a'}} \widehat{p}_{i|S_{a'}} \cdot m_{a' \to i}^{(t-1)} \,,$$
$$m_{a \to i}^{(t)} = \sum_{i' \in S_a} m_{i' \to a}^{(t)} \,.$$

Now, suppose that the empirical edge weights $\widehat{p}_{i|S_a}$ are equal to the true weights $p_{i|S_a} = \frac{w_i}{\sum_{j \in S_a} w_j}$, $\forall i \in [n], a \in [d]$. Also, suppose on some iteration $t \geq 1$, the item messages $m_{i \to a}^{(t)}$ become equal to the item weights w_i, $\forall i \in [n]$. Then it is easy to observe that the next iteration of messages $m_{i \to a}^{(t+1)}$ are also equal to w_i. Therefore, the true weights $\mathbf{w}$, in some sense, are a fixed point of the above set of equations. The following lemma shows that the ASR algorithm is equivalent to this message passing algorithm.

Lemma 4.6.1. *For any realization of choice data $\mathbf{Y}$, there is a one-to-one correspondence d each iteration of the message passing algorithm (6) and the corresponding power iteration of the ASR algorithm (5), and both algorithms return the same estimates $\widehat{\mathbf{w}}$ for any $\mathbf{Y}$.*

Proof. In the message passing algorithm, the item to set messages $m_{i \to a}^{(r)}$ in round r correspond to the estimates of the item weights. One can verify that the estimate $\widehat{w}^{(r)}$ of item i in round

r evolves according to the following equation.

$$\widehat{w}_i^{(r+1)} = \frac{1}{d_i} \sum_{a:i \in S_a} p_{i|S_a} \cdot \sum_{j \in S_a} \widehat{w}_j^{(r)} \,.$$

We can represent this system of equations compactly using the following matrices. Let $\widehat{\mathcal{V}} \in \mathbb{R}^{d \times n}$ be a matrix such that

$$\widehat{V}_{ai} := \begin{cases} \frac{p_{i|S_a}}{d_i} & \text{if } (i,a) \in E \\ 0 & \text{otherwise} \end{cases} \,, \tag{4.6.1}$$

and $\mathbf{B} \in \mathbb{R}^{n \times d}$ be a matrix such that

$$B_{ia} := \begin{cases} 1 & \text{if } (i,a) \in E \\ 0 & \text{otherwise} \end{cases} \,, \tag{4.6.2}$$

Thus, we can represent the weight update from round (r) to round $(r+1)$ as

$$\begin{aligned} \widehat{\mathbf{w}}^{(r+1)} &= (\mathbf{B}\widehat{\mathcal{V}})^\top \widehat{\mathbf{w}}^{(r)} = \widehat{\mathbf{M}}^\top \widehat{\mathbf{w}}^{(r)} \\ &= (\widehat{\mathbf{M}}^\top)^r \widehat{\mathbf{w}}^{(0)} \,, \end{aligned}$$

where $\widehat{\mathbf{M}} := \mathbf{B}\widehat{\mathcal{V}}$, with entry (i,j) of $\widehat{\mathbf{M}}$ being

$$\widehat{M}_{ij} := \frac{1}{d_j} \sum_{a:i,j \in S_a} p_{j|S_a} \,. \tag{4.6.3}$$

The above equation implies that the message passing algorithm is essentially a power iteration on the matrix $\widehat{\mathbf{M}}$. Now, it is easy to see that $\widehat{\mathbf{M}} = \mathcal{D}\widehat{\mathbf{P}}\mathcal{D}^{-1}$ where $\widehat{\mathbf{P}}$ is the transition matrix constructed by ASR (*Eq.* (4.3.2)). Therefore, there is a one-to-one correspondence between the power iterations on $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{P}}$. More formally, if we initialize with $\widehat{\mathbf{w}}^{(0)}$ in the power iteration on $\widehat{\mathbf{M}}$, and initialize with $\widehat{\boldsymbol{\pi}}^{(0)} = \mathcal{D}\widehat{\mathbf{w}}^{(0)}$ in the power iteration on $\mathbf{P}$, then the iterates at the r-th step will be related as $\widehat{\boldsymbol{\pi}}^{(r)} = \mathcal{D}\widehat{\mathbf{w}}^{(r)}$. Furthermore, if $\widehat{\boldsymbol{\pi}}$ is the stationary

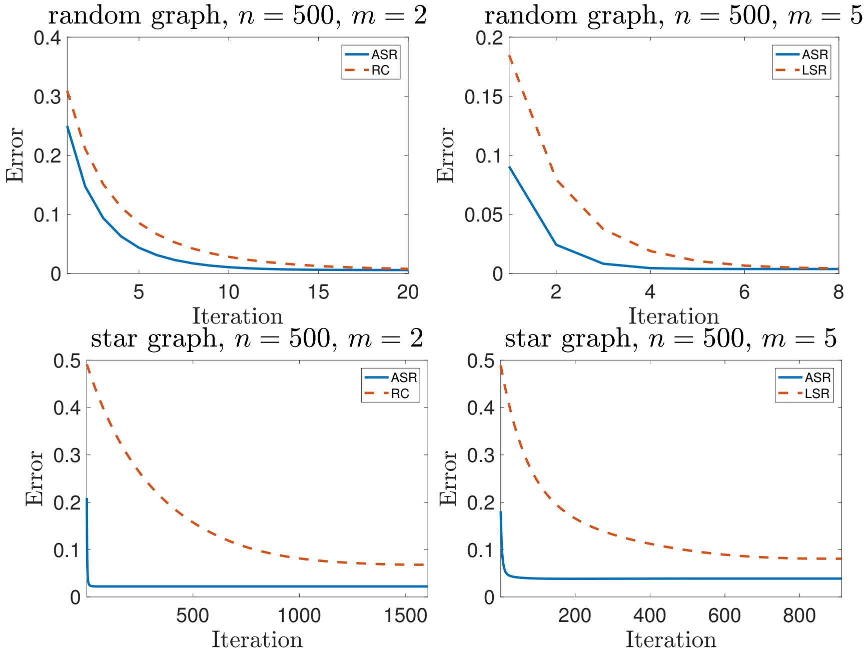


Figure 8: Results on synthetic data: L_1 error vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for $m = 2$) and the LSR algorithm (for $m = 5$), on data generated from the MNL/BTL model with the random and star graph topologies.

distribution of $\widehat{\mathbf{P}}$, then $\widehat{\mathbf{w}} = \mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}$ is the corresponding dominant left eigenvector of $\widehat{\mathbf{M}}$, i.e. $\mathcal{D}^{-1}\widehat{\boldsymbol{\pi}} = \widehat{\mathbf{M}}^\top \mathcal{D}^{-1}\widehat{\boldsymbol{\pi}}$. Also, $\widehat{\mathbf{w}}$ is exactly the estimate (after normalization) returned by both the ASR and the message passing algorithm upon convergence. Thus, we can conclude that the message passing algorithm is identical to ASR for any realization of comparison data generated according to the MNL model. □

The above lemma gives an interesting connection between spectral ranking under the MNL model and message passing/belief propagation. Such connections have been observed for other problem such as the problem of aggregating crowdsourced binary tasks (Khetan and Oh, 2016b). A consequence of this connection is that it facilitates a fully decentralized distributed implementation of the ASR algorithm. This can be very useful for modern applications, where machines can communicate local parameter updates to each other, without explicitly communicating the data.

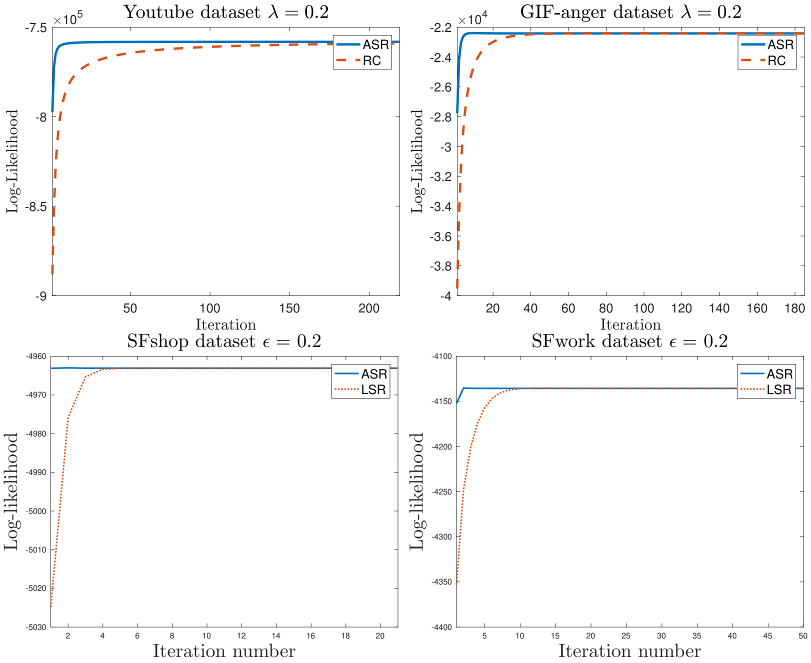


Figure 9: Results on real data: Log-likelihood vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for pairwise choice data) and the LSR algorithm (for multi-way choice data), all with regularization parameter set to 0.2.

4.7 Experiments

In this section we perform experiments on both synthetic and real data to compare our algorithm to the existing LSR (Maystre and Grossglauser, 2015) and RC (Negahban et al., 2017) algorithms for recovering the weight vector $\mathbf{w}$ under the MNL and BTL model, respectively. The implementation[4] of our algorithm is based on applying the power method on $\widehat{\mathbf{P}}$ (Eq. (4.3.2)). The power method was chosen due to its simplicity, efficiency, and scalability to large problem sizes. Similarly, the implementations of LSR and RC are based on applying the power method on $\widehat{\mathbf{P}}^{\text{LSR}}$ (Eq. (4.4.2)), and $\widehat{\mathbf{P}}^{\text{RC}}$ (Eq. (4.4.1)), respectively. In the definition of $\widehat{\mathbf{P}}^{\text{LSR}}$, the parameter ϵ was chosen to be the maximum possible value that ensures $\widehat{\mathbf{P}}^{\text{LSR}}$ is a Markov chain.

[4] code available: https://github.com/agarpit/asr

4.7.1 Synthetic Data

We conducted experiments on synthetic data generated according to the MNL model, with weight vectors $\mathbf{w}$ generated randomly (details below). We compared our algorithm with the LSR algorithm for choice sets of size $m = 5$, and with the RC algorithm for sets of size $m = 2$. We used two different graph topologies for generating the comparison graph G_c, or equivalently the choice sets:

1. **Random Topology:** This graph topology corresponds to random graphs where $n \log_2(n)$ choice sets are chosen uniformly at random from all the $\binom{n}{m}$ unique sets of cardinality m. This topology is very close to the Erdős-Rényi topology which has been well-studied in the literature. In fact the degree distributions of nodes in this random topology are very close to the degree distributions in the Erdős-Rényi topology (Mezard and Montanari, 2009). The only reason we study the former is computational, as iterating over all $\binom{n}{m}$ hyper-edges is computationally challenging.

2. **Star Topology:** In this graph topology, there is a single item that belongs to all sets; the remaining $(m-1)$ items in each set are contained only in that set. We study this topology because it corresponds to the choice sets used in Example 4.4.6, where there was a factor of $\Omega(n)$ gap in the spectral gap between our algorithm and the other algorithms.

In our experiments we selected $n = 500$[5], and the weight w_i of each item $i \in [n]$ was drawn uniformly at random from the range $(0, 1)$; the weights were then normalized so they sum to 1. A comparison graph G_c was generated according to each of the graph topologies above. The parameter L was set to $300 \log_2 n$. The winner for each choice set was drawn according to the MNL model with weights $\mathbf{w}$. The convergence criterion for all algorithms was the same: we run the algorithm until the L_1 distance between the new estimates and the old estimates is ≤ 0.0001. Each experiment was repeated 100 times and the average values over all trials are reported. For $n = 500$, $m \in \{2, 5\}$, and both graph topologies described

[5] Results for other values of n are given in the Appendix.

Table 2: Statistics for real world datasets

Dataset	n	m	d	$d_{\max}/d_{\min}$
Youtube	21207	2	394007	600
GIF-anger	6119	2	64830	106
SFwork	6	3-6	12	4.3
SFshop	8	4-8	10	1.9

above, we compared the convergence as a function of the number of iterations[6] for each algorithm. We plotted the L_1 error of the estimates produced by these algorithms after each iteration. The plots are given in Figure 8. These plots verify the mixing time analysis of Section 4.4, and show that our algorithm converges much faster than RC and LSR, and orders of magnitude faster in the case of the star graph.

4.7.2 Real World Datasets

We conducted experiments on the YouTube dataset (Shetty, 2012), GIF-anger dataset (Rich et al.), and the SFwork and SFshop (Koppelman and Bhat, 2006) datasets. Table 2 gives some statistics about these datasets. We also plot the degree distributions of these datasets in the Appendix. For these datasets, a ground-truth $\mathbf{w}$ is either unknown or undefined; and hence, we compare our algorithm and the RC/LSR algorithm with respect to the log-likelihood of the estimates as a function of number of iterations. Due to the number of comparisons per set (or pair) being very small, in order to ensure irreducibility of random walks, we use a regularized version of all algorithms (see Appendix, and also Section 3.3 in Negahban et al. (2017), for more details). Here, we give results when the regularization parameter λ is set to 0.2, and defer the results for other parameter values to the Appendix. The results are given in Figure 9. We observe that our algorithm converges rapidly to the peak log-likelihood value while RC and LSR are always slower in converging to this value.

[6] We also plotted the convergence as a function of the running time; the results were similar as the running time of each iteration is similar for all these algorithm.

4.8 Conclusion

We presented a spectral algorithm for learning parameters of the MNL/BTL model from pairwise/multiway choices. Our algorithm is considerably faster than previous algorithms; in addition, our analysis yields improved sample complexity results for estimation under the BTL and MNL model. We also give a message passing/belief propagation interpretation for our algorithm. In the future it would be interesting to see if one can use our algorithm to give better guarantees for recovery of top-k items under MNL. Moreover, it would also be interesting to study learning algorithms for other choice models such as multinomial probit model (MNP), nested logit model, and mixture of MNLs etc.

Chapter 5

Multiarmed Bandits and Discrete Choice Models

In the previous chapter we designed an algorithm for learning the parameters of the multinomial logit (MNL) model from offline choice datasets. In this chapter we will continue our discussion at the interface of machine learning and choice modeling and design algorithms for learning under different choice models in the online multi-armed bandit setting.

5.1 Introduction

5.1.1 Background

As discussed in the previous chapter, discrete choice models have gained a lot of interest in machine learning due to the onset of online services in domains including entertainment and shopping, that use machine learning to recommend alternatives to users and help them make better choices. In the previous chapter our goal was to learn a choice model from offline choice data collected over time. However, in a lot of applications, the interaction of users with the learning algorithm happens in an online manner, i.e. in sequential rounds of interaction. Hence, it is desirable for these recommendation algorithms to continuously learn about the tastes/choices of these users from this sequential interaction and recommend better set of products progressively over time.

A widely studied setting for online learning is the *multi-armed bandits* setting where the learner interacts with the environment in a sequential manner, and each time collects partial feedback which is used to improve the interaction over time by minimizing an appropriately notion of *regret*. Motivated by applications in online recommendation systems and advertising, we seek to study choice models under this setting of online multi-armed bandits.

Previously, Yue et al. (2009) introduced the framework of *dueling bandits* that studies *pairwise* choice models under the multi-armed bandits setting. This framework has gained a lot of interest in machine learning in recent years (Yue et al., 2009; Yue and Joachims, 2011; Yue et al., 2012; Urvoy et al., 2013; Ailon et al., 2014; Zoghi et al., 2014, 2015a,b; Dudik et al., 2015; Jamieson et al., 2015; Komiyama et al., 2015a, 2016; Ramamohan et al., 2016; Chen and Frazier, 2017). Here there are n arms $\{1, \ldots, n\}$; on each trial t, the learner pulls a pair of arms (i_t, j_t), and receives *pairwise choice* indicating which of the two arms has a better quality/reward. In the regret minimization setting, the goal is to identify the 'best' arm(s) while also minimizing the regret due to playing sub-optimal arms in the learning (exploration) phase.

In many applications, however, it can be natural for the learner to pull more than two arms at a time, and seek relative feedback among them. For example, in recommender systems, it is natural to display several items or products to a user, and seek feedback on the most preferred item among those shown. In online advertising, it is natural to display several ads at a time, and observe which of them is clicked (preferred). In online ranker evaluation for information retrieval, one can easily imagine a generalization of the setting studied by Yue and Joachims (2009), where one may want to "multi-leave" several rankers at a time to help identify the best ranking system while also presenting good/acceptable results to users using the system during the exploration phase. In general, there is also support in the marketing literature for showing customers more than two items at a time (Johnson et al., 2012). Motivated by these applications, we seek to move beyond the pairwise choice setting of dueling bandits and design a new framework that can incorporate more general multiway choices.

5.1.2 Our Contributions

We introduce a framework that generalizes the dueling bandit problem to allow the learner to pull more than two arms at a time. Here, on each trial t, the learner pulls a set S_t of up to k arms (for fixed $k \in \{2, \ldots, n\}$), and receives relative feedback in the form of a multiway choice $y_t \in S_t$ indicating which arm in the set has the highest quality/reward. The goal of the learner is again to identify a 'best' arm (to be formalized below) while minimizing a suitable notion of regret that penalizes the learner for playing sub-optimal arms during the exploration phase. We term the resulting framework *choice bandits*.

In the (stochastic) dueling bandits framework, the underlying probabilistic model from which feedback is observed is a *pairwise comparison model*, which for each pair of arms (i, j), defines a probability P_{ij} that arm i has higher reward/quality than arm j. In our choice bandits framework, the underlying probabilistic model is a *multiway choice model*, which for each set of arms $S \subseteq [n]$ with $|S| \leq k$ and each arm $i \in S$, defines a probability $P_{i|S}$ that arm i has the highest reward/quality in the set S. Figure 10 gives the hierarchy of choice models considered in this chapter.

We first consider choice bandits under the well-known *multinomial logit* (MNL) choice model (Luce, 1959; Plackett, 1975; McFadden, 1974), which generalizes the Bradley-Terry-Luce (BTL) model for pairwise comparisons (Bradley and Terry, 1952b; Luce, 1959). Under this model, each arm i is associated with a weight $w_i > 0$, and the choice probabilities are given by $P_{i|S} = w_i / \sum_{j \in S} w_j$. We design a computationally efficient algorithm, which we term *Winner Beats All – Lazy* (WBA-L), that achieves an instance-wise optimal regret bound of $O\big(n \log n \log T\big)$, where T is the number of trials (horizon). This bound significantly improves upon the worst-case $O(n^2 + n \log T)$ bound achieved by the recent MaxMinUCB algorithm designed for the MNL model (Saha and Gopalan, 2019a).

We then study choice bandits under a new class of choice models, that are characterized by the existence of a unique *generalized Condorcet winner* (GCW), which we define to be an arm that has larger probability of being chosen than any other arm in any choice set. This class includes as special cases the multinomial logit (MNL) and multinomial probit (MNP) (Thurstone, 1927) choice models, and more generally, the class of random utility models with i.i.d. noise (IID-RUMs) (Marschak, 1960; Domencich and McFadden, 1975). The main contribution of this work is to design a computationally efficient algorithm, which we term *Winner Beats All – Aggressive* (WBA-A), that achieves an instance-wise asymptotically optimal regret bound of $O\big(n^2 \log n + n \log T\big)$ under this large class of choice models that

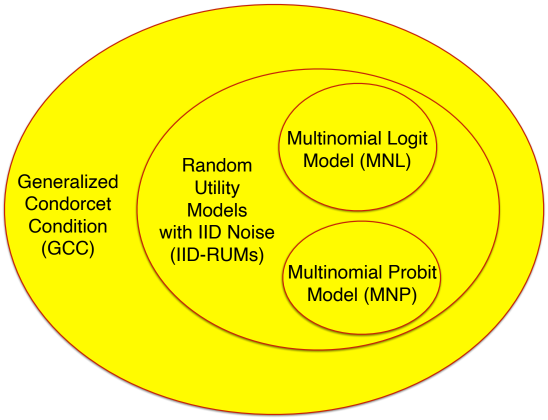

Generalized Condorcet Condition (GCC)
Random Utility Models with IID Noise (IID-RUMs)
Multinomial Logit Model (MNL)
Multinomial Probit Model (MNP)

Figure 10: The hierarchy of choice models considered in this work.

exhibit a unique GCW.

The main challenge in designing an algorithm under our framework is that the space of exploration (number of possible sets the learner can play) is $\Theta(n^k)$ which is large even for moderate k. Therefore, it can be challenging to simultaneously *explore/learn* the choice sets with low regret out of the possible $\Theta(n^k)$ sets and *exploit* these low regret sets. We overcome these challenges by extracting just $O(n^2)$ pairwise statistics from the observed multiway choices under different sets, and using these statistics to find choice sets with low regret. Since these pairwise statistics are extracted from multiway choices under different sets, a technical challenge is to show that these statistics are concentrated. We resolve this challenge by using a novel coupling argument that couples the stochastic process generating choices with another stochastic process, and showing that pairwise estimates according to this other process are concentrated. We believe that our results for efficient learning under this large class of choice models that is considerably more general than the MNL class are of independent interest.

We also run experiments on several synthetic and real-world datasets. Our experiments on these datasets show that our algorithms for the special case of $k = 2$ are competitive as compared to previous dueling bandit algorithms, even though they are designed for a more general setting. For the case of $k > 2$, we compare our algorithms with the MaxMinUCB algorithm of Saha and Gopalan (2019a) which was designed for the MNL model. We observe that our algorithms perform better in terms of regret than MaxMinUCB under all datasets (even under synthetic MNL datasets). We further observe that under several datasets the regret achieved by our algorithms for $k > 2$ is better than the regret for $k = 2$.

The following is a summary of our contributions

1. **Modeling Contributions:** We formalize a new framework that generalizes the dueling bandit framework by allowing the learner to play larger choice sets. Our framework opens up several new questions, including the possibility of designing algorithms for specific types of choice models of interest in various applications. We also propose a new non-parametric class of choice models (GCC) which include several well-studied choice models such as MNL, MNP, and more generally all IID-RUMs as special cases, and can be of independent interest in other multiway choice settings such as dynamic assortment optimization (Sauré and Zeevi, 2013).

2. **Algorithmic Contributions:** We develop a novel algorithmic framework for extracting pairwise statistics from multiway choices and making decisions based on these pairwise statistics. This allows the learner to have the flexibility of playing larger choice sets while being computationally efficient and achieving tight regret under a wide range of choice models.

3. **Technical Contributions:** We believe that our results for learning this large GCC class of choice models are of independent interest. Of particular interest are our ideas of extracting and aggregating potentially inconsistent pairwise preferences from multiway choices, and our concentration results used to establish confidence interval bounds on these preference estimates.

5.1.3 Related work.

There has been a lot of work recently in bandit settings where more than two arms are played at once (although no previous work considers choice models at the level of generality we do).

Table 3: Overview of related work in regret minimization settings. There are several definitions of 'best' arm; the reader is encouraged to refer to the relevant papers and to our problem setting for details. (Note: in multi-dueling bandits, $\emptyset$ denotes no feedback; in stochastic click bandits, O_t denotes an ordered set; in combinatorial bandits, $\mathcal{S}$ denotes a set of allowed subsets; in dynamic assortment optimization, 0 denotes the "no-purchase" option.)

Problem	**Arms Pulled in Round t**	**Feedback in Round t**	**Goal**
Dueling Bandits	$(i_t, j_t) \in [n]^2$	$y_t \in \{i_t, j_t\}$	Min. regret w.r.t. best arm
Multi-dueling Bandits	$S_t \in [n]^k$	$Y_t {=} \{0, 1, \emptyset\}^{k \times k}$	Min. regret w.r.t. best arm
Combinatorial Bandits	$S_t \in \mathcal{S} \subseteq 2^{[n]}{:}\lvert S_t\rvert \leq k$	$y_t(i) \in \mathbb{R}\ \forall i \in S_t$	Min. regret w.r.t. top-k arms
Combinatorial Bandits with Relative Feedback	$S_t \subseteq [n]{:}\lvert S_t\rvert \leq k$	$O_t \subseteq S_t, \lvert O_t\rvert \leq m$	Min. regret w.r.t. best arm (MNL)
Battling Bandits	$S_t \in [n]^k$	$y_t \in S_t$	Min. regret w.r.t. best arm (PS)
Stochastic Click Bandits	$O_t \subseteq [n]{:}\lvert O_t\rvert = k,$	$y_t \subseteq O_t$	Max. expected clicks clicks
Dynamic Assortment	$\{0\} \cup S_t \subseteq [n]{:}\lvert S_t\rvert \leq k$	$y_t \in S_t$	Max. expected revenue
Choice Bandits	$S_t \subseteq [n]{:}\lvert S_t\rvert \leq k$	$y_t \in S_t$	Min. regret w.r.t. best arm

We briefly review related work here; see also Table 3.

Multi-dueling bandits. In *multi-dueling bandits* (Brost et al., 2016; Schuth et al., 2016; Sui et al., 2017), the learner pulls a set S_t of k items; however, the feedback received by the learner is assumed to be drawn from a pairwise comparison model (in particular, the learner observes some subset of the $\binom{k}{2}$ possible pairwise comparisons among items in S_t). In contrast, in our choice bandits setting, the learner receives the outcome of a direct multiway choice among the items in S_t, generated from a multiway choice model.

Combinatorial bandits. In *combinatorial (semi) bandits* (Gai et al., 2012; Chen et al., 2013; Kveton et al., 2015; Combes et al., 2015), each arm i is associated with an unknown random variable (stochastic reward) Y_i; the learner pulls a set S_t of up to k arms (possibly from some set of 'allowed' sets $\mathcal{S} \subseteq 2^{[n]}$), and observes the realized rewards $y_t(i)$ for all arms i in S_t. The goal is to maximize the cumulative sum of all rewards. This is different from our choice bandits setting; in our setting, the learner observes only which arm is chosen from the set S_t of arms pulled, rather than any absolute reward feedback (indeed, in our setting, arms may not be associated with individual rewards at all).

Combinatorial bandits with relative feedback. In this very recent framework Saha and Gopalan (2019a), the learner pulls a set S_t of up to k arms, and observes top-m ordered feedback drawn according to the MNL model, for some $m \leq k$. In contrast, we only observe the (top-1) choice feedback from the set S_t that is played. Moreover, we study a much more general class of choice models than the MNL model studied by them. For the special case of (top-1) choice feedback under MNL, we give better algorithms with (almost) optimal instance-wise bounds as compared to their MaxMinUCB algorithm which has a worst-case bound.

Stochastic click bandits. In *stochastic click bandits* (Zoghi et al., 2017), the learner pulls an *ordered* set of k arms/documents, and observes *clicks* on a subset of these documents, drawn according to an underlying *click model* which is a probabilistic model for click generation over an ordered set. However, click models in their setting are different than choice models in our setting, and neither can be cast as a special case of the other.

Battling bandits. Another related setting is that of *battling bandits* (Saha and Gopalan, 2018), where the learner pulls a set S_t of *exactly* k arms and receives a feedback indicating which arm was chosen. However, their setting considers a specific pairwise-subset (PS) choice model that is defined in terms of a pairwise comparison model, whereas we consider much more general choice models.

Preselection bandits. There has been a recent framework called *preselection bandits* Bengs and Hüllermeier (2019) where two settings are considered: (1) where the learner pulls a set S_t of size *exactly* k, (2) where the learner pulls a set S_t of any size less than n. In both settings the learner receives feedback drawn from the MNL model. Firstly, the two settings considered by this work are different than our setting where the learner plays a set of size up to k. Secondly, we study a much more general class of choice models than the MNL model studied by them.

Dynamic assortment optimization. In *dynamic assortment optimization* Rusmevichientong et al. (2010); Sauré and Zeevi (2013); Agrawal et al. (2016, 2017); Chen and Wang (2017), there are n products and each product is associated with a revenue. The learner plays an assortment S_t of up to k products, and observes a feedback indicating which (if any) of the products was purchased; the goal of the learner is to maximize the expected revenue.

Best-of-k bandits (PAC setting). Simchowitz et al. (2016) consider a *best-of-k bandits* setting, where again the learner pulls a set S_t of k arms; however here each arm i is associated with an unknown random variable (stochastic reward) Y_i. Of the various types of feedback that are considered, the *marked bandit* feedback corresponds to a setting that is similar to our choice bandits framework, however, the analysis in Simchowitz et al. (2016) is in the PAC/pure exploration setting, while ours is in the regret minimization setting.

Top-k identification under MNL model (PAC setting). Recently, there has also been work on identifying the top-k items under an MNL model from actively selected sets S_t in the PAC/pure exploration setting Chen et al. (2018).

5.1.4 Organization

We set up the choice bandits problem in Section 5.2. We present a fundamental lower bound for our choice bandits problem in Section 5.3. We present our two algorithms in Section 5.4. We present regret upper bounds for our algorithms in Section 5.5. We present experimental results on synthetic and real world datasets in Section 5.6. We present proofs of our theoretical results in Section 5.7. We finally conclude with a brief discussion in Section 5.8.

5.2 Problem Setup and Preliminaries

In the choice bandits problem, there are n arms $[n] := \{1, \dots, n\}$, and a set size parameter $2 \leq k \leq n$. On each trial t, the learner pulls (selects/plays) a choice set $S_t \subseteq [n]$ of up to k arms, i.e. with $|S_t| \leq k$, and receives as feedback $y_t \in S_t$, indicating the arm that is most preferred in S_t. We assume the feedback y_t is generated probabilistically from an underlying *multiway choice model*, which defines for each $S \subseteq [n]$ such that $|S| \leq k$, and arm $i \in S$, a *choice probability* $P_{i|S}$ which corresponds to the probability that arm i is the most preferred arm in S.[1] Before defining appropriate notions of 'best' arm and regret for the learner, we give some examples of multiway choice models.

5.2.1 Random Utility Models with IID Noise (IID-RUMs)

IID-RUMs are a well-known class of choice models that have origins in the econometrics and marketing literature (Marschak, 1960; Train, 2003). Under an IID-RUM, the (random) utility associated with arm $i \in [n]$ is given by $U_i = v_i + \epsilon_i$ where $v_i \in \mathbb{R}$ is a deterministic utility and the $\epsilon_i \in \mathbb{R}$ are the random noise variables drawn i.i.d. from a distribution $\mathcal{D}$ over reals. For a set S, the probability of choosing $i \in S$ is given by

$$P_{i|S} = \Pr\left(U_i > U_j \, \forall j \in S \setminus \{i\}\right).$$

[1] Note that for the special case of $k = 2$, our framework reduces to dueling bandits; the pairwise comparison probabilities $P_{ij} := \Pr\left(i \succ j\right)$ in dueling bandits can be viewed as pairwise choice probabilities $P_{i|\{i,j\}}$.

We will sometimes also refer to v_i as the weight of item i. Under any IID-RUM, if $v_i > v_j$ for some $i, j \in [n]$, then arm i will be more likely to be chosen than arm j in any set. The IID-RUM class contains some popular models, such as the multinomial logit (MNL) (Luce, 1959; Plackett, 1975; McFadden, 1974) and multinomial probit (MNP) (Thurstone, 1927), as special cases.

Example 5.2.1 (MNL)**.** *Under MNL, the noise distribution $\mathcal{D}$ is Gumbel$(0, 1)$ and the probability $P_{i|S}$ of choosing an item i from a set S has the following closed form expression:*

$$P_{i|S} := \frac{e^{v_i}}{\sum_{j \in S} e^{v_j}} \,.$$

It is clear from this expression that arms with higher weights are more likely to be chosen.

Example 5.2.2 (MNP)**.** *Under the MNP model, the noise distribution $\mathcal{D}$ is the standard Normal distribution $\mathcal{N}(0, 1)$. Unlike MNL, there is no closed form expression for the choice probabilities.*

Under IID-RUMs there is a clear notion of 'best' arm: an arm that has the highest weight $\max_{i \in [n]} v_i$. We now define a strictly more general class of models where there is a clear notion of 'best' arm.

5.2.2 A New Class of Choice Models

We introduce a new class of multiway choice models that are characterized by the following condition that requires the existence of a unique 'best' arm.

Definition 5.2.3 (Generalized Condorcet Condition (GCC))**.** *A choice model is said to satisfy the GCC condition if there exists a unique arm $i^* \in [n]$ such that for every choice set $S \subseteq [n]$ that contains i^*, we have $P_{i^*|S} > P_{j|S}$ for all $j \in S \setminus \{i^*\}$.*

Intuitively, the above condition requires the existence of a unique arm that is always (stochastically) preferred to all other arms, no matter what other arms are shown with it. This condition is a generalization of the Condorcet condition studied for pairwise comparison models (Zoghi et al., 2014; Komiyama et al., 2015a). Just as the Condorcet condition need not be satisfied for all pairwise comparison models, similarly, GCC need not be satisfied by all multiway choice models. Below we show that the GCC condition is satisfied for all IID-RUMs subject to a minor technical condition.

Lemma 5.2.4 (IID-RUMs satisfy GCC)**.** *For any IID-RUM choice model with utility for arm $i \in [n]$ given by $U_i = v_i + \epsilon_i$, the GCC condition is satisfied if $|\operatorname{argmax}_{i \in [n]} v_i| = 1$.*

In this work, we study the class of all choice models where the GCC is satisfied. Under GCC, we will refer to the unique 'best' arm as the generalized Condorcet winner (GCW) and denote it by i^*. Note that for any set S containing the GCW i^*, we must have $P_{i^*|S} \geq \frac{1}{|S|}$.

5.2.3 Regret Notion

Similar to dueling bandits, the goal of the learner in our setting is to identify the best arm while also playing good/competitive sets with respect to this arm during the exploration phase.[2] Hence, our notion of regret measures the sub-optimality of a choice set S relative to i^*, and is a generalization of the regret defined by Saha and Gopalan (2019a) for the special case of MNL choice models. Moreover, under our notion of regret it is optimal to play $S^* = \{i^*\}$, i.e. regret of playing S^* is 0. The regret of a set is defined to be the sum of regret due to individual arms in the set, and the regret for an arm corresponds to the 'margin' by which the best arm i^* beats this arm. In other words, the regret of an arm corresponds to the *shortfall in preference probability* due to pulling this arm over the 'best' arm.

Definition 5.2.5 (Regret)**.** *The regret $r(S)$ for $S \subseteq [n]$ is defined as: $r(S) := \sum_{i \in S} \big(P_{i^*|S \cup \{i^*\}} - P_{i|S \cup \{i^*\}}\big)$.*

This notion of regret can be interpreted as: $r(S)$ is the sum over all arms $i \in S$, the fraction of consumers that will choose i^* minus the fraction of consumers that will choose i when i^* is played together with S. It is easy to see that $r(\{i^*\}) = 0$, and $0 \leq r(S) \leq |S|$ for any set $S \subseteq [n]$.

[2] Note that we are not working in the pure exploration setting, where all sets are of equal cost during exploration.

Example 5.2.6 (Linearly growing regret)**.** *Consider a choice model where arm* 1 *is the GCW, and for each set* S *containing arm 1, we have* $P_{1|S} = 0.51$ *and* $P_{i|S} = \frac{0.49}{|S|-1}$ $\forall i \in S \backslash \{1\}$*. Then* $r(\{1, \ldots, m\}) = 0.51 \times (m-1) - 0.49$*.*

In the above example, the regret increases linearly as we increase m. The following gives an example where the arms are much more 'competitive' and regret is smaller.

Example 5.2.7 (Sub-linearly growing regret)**.** *Consider the MNL choice model with weights* $v_1 = \log(1+\epsilon)$*, for* $\epsilon > 0$*, and* $v_2 = \cdots = v_n = 0$*. Then* $r(\{1, \cdots, m\}) = \sum_{i \in [m]} \frac{e^{v_1} - e^{v_i}}{\sum_{j \in [m]} e^{v_j}} = \frac{\epsilon(m-1)}{m+\epsilon}$*.*

The regret here increases much more slowly in terms of m. Note that our regret is not necessarily well-defined in the dueling bandits setting, due to the need to consider choice probabilities for sets of size 3 even when one plays only sets of size 2.

Under the above notion of regret, the goal of an algorithm $\mathcal{A}$ is to minimize its cumulative regret over T trials, defined as: $R(T) = \sum_{t=1}^{T} r(S_t)$.

5.3 A Fundamental Lower Bound

In this section we present a regret lower bound for our choice bandits problem. We say that an algorithm is *strongly consistent* under GCC if its expected regret over T trials is $o(T^a)$ for any $a > 0$ under any model in this class. Given a GCC choice model and an arm $i \neq i^*$, let us define the gap parameter Δ_{i^*i} as

$$\Delta_{i^*i} := \min_{S \subseteq [n]: |S| \leq k \text{ and } i, i^* \in S} \frac{P_{i^*|S} - P_{i|S}}{P_{i^*|S} + P_{i|S}} \,. \tag{5.3.1}$$

The following theorem presents a lower bound for any strongly consistent algorithm in terms of these gap parameters.

Theorem 5.3.1. *Given a set of arms* $[n]$*, choice set size bound* $k \leq n$*, there exist GCC choice models such that when choice outcomes are drawn according to these models, the regret* *incurred by any algorithm $\mathcal{A}$ that is strongly consistent under GCC is lower bounded as:*

$$\liminf_{T\to\infty} \frac{\mathbf{E}\left[R(T)\right]}{\log T} = \Omega\left(\sum_{i\in[n]\setminus\{i^*\}} \frac{1}{\Delta_{i^*i}}\right),$$

where T is the time-horizon. Moreover, if the underlying model is MNL with parameters $v_1, v_2, \cdots v_n \in \mathbb{R}$, then:

$$\liminf_{T\to\infty} \frac{\mathbf{E}\left[R(T)\right]}{\log T} = \Omega\left(\sum_{i\in[n]\setminus\{i^*\}} \frac{1}{\Delta^{\text{MNL}}_{i^*i}}\right)$$

where $\Delta^{\text{MNL}}_{i^*i} = \frac{e^{v_{i^*}} - e^{v_i}}{e^{v_{i^*}} + e^{v_i}}$, for $i \in [n] \setminus \{i^*\}$.

Discussion. The above bound shows that any algorithm for the choice bandits problem needs to incur instance-dependent $\Omega(n \log T)$ regret in the worst case. Note that the above lower bound does not depend on the choice set size parameter k. If the choices are generated from an underlying MNL model, then the above theorem gives an instance-dependent lower bound for the regret of any algorithm. Note that Saha and Gopalan (2019a) also provided a lower bound under MNL for our notion of regret, however, their bound depends on the worst-case gap between i^* and any other arm $i \neq i^*$, while we provide a more fine-grained bound under MNL which depends on gaps between i^* and each individual arm $i \in [n]$.

In order to prove the above bound we construct a pair of instances that have different GCW arms, and use the information divergence lemma of Kaufmann et al. (2016) in order to characterize the minimum number of samples needed in order to collect the 'information' needed to separate these two instances. We provide a full proof of this lower bound in Section 5.7.1.

5.4 Algorithms

In this section we describe our two algorithms, termed *Winner Beats All – Aggressive* (WBA-A) and *Winner Beats All – Lazy* (WBA-L). The WBA-L algorithm is designed for the MNL model while the WBA-A algorithm is designed for the more general GCC class

Algorithm 7 Winner Beats All – Aggressive (WBA-A)

```
Input: set of arms $[n]$, size of choice set $k$, parameter $C$
$t \leftarrow 1$, $r \leftarrow 1$, $A_r \leftarrow [n]$, $a_t \leftarrow \text{Unif}([n])$, $Q \leftarrow \emptyset$
$\widehat{P}_{ij} \leftarrow \frac{1}{2}$, $\forall i, j \in [n]$
while $t \leq T$ do
    Select largest $S \subseteq A_r \setminus \{Q \cup a_t\}$ with $|S| \leq k-1$ and $\widehat{P}_{ia_t} \leq \frac{1}{2}$, $\forall i \in S$
    Let $S_t \leftarrow S \cup \{a_t\}$; while $|S_t| < k$ and $A_r \setminus S_t \neq \emptyset$: add an (arbitrary) arm from $A_r \setminus S_t$ to $S_t$
    Play set $S_t$ and receive $y_t \in S_t$ as feedback; $Q \leftarrow Q \cup S$
    For all $i \in S_t$, calculate $\widehat{P}_{ia_t}(t)$ and $\mathcal{J}_i(t, C)$
    if $\forall\ i \in A_r \setminus \{Q \cup a_t\}$, $\widehat{P}_{ia_t}(t) > \frac{1}{2}$ then
        $a_{t+1} \leftarrow \text{argmax}_{i \in [n]} \sum_{j \in [n] \setminus Q} \mathbb{1}[\widehat{P}_{ji}(t) \leq \frac{1}{2}]$
    else
        $a_{t+1} \leftarrow a_t$
    end if
    if $Q = A_r$ or $S = \emptyset$ then
        $A_{r+1} \leftarrow \emptyset$, $r \leftarrow r + 1$
        for $i \in [n]$ do
            if $\mathcal{J}_i(t, C) = 0$, then $A_r \leftarrow A_r \cup \{i\}$
        end for
        $a_{t+1} \leftarrow \text{argmax}_{i \in [n]} \sum_{j \in [n]} \mathbb{1}[\widehat{P}_{ji}(t) \leq \frac{1}{2}]$, $Q \leftarrow \emptyset$
    end if
    $t \leftarrow t + 1$
end while
```

of models. However, the two algorithms are built upon the common principle of quickly isolating the best arm i^* by using the fact that this arm stochastically beats all other arms in any choice set.

Both our algorithms divide their execution into rounds and each round can contain up to n trials depending on problem parameters and the execution history. We will use r as an index for a round, and t as a (global) index for a trial. For each round r, both algorithms maintain a set A_r of active arms. These are a set of arms for which the algorithm is still not confident enough that these are 'bad' arms. Note that an arm that is inactive in a particular round, can become active in a later round. We also maintain a set Q that is initialized to being empty at the beginning of each round and keeps track of the arms in A_r that have been played so far in the round.

Given a trial t that falls in round r, both algorithms first select a set $S \subseteq A_r \setminus Q$ (arbitrarily) of up to $k-1$ arms in A_r that have not been played so far in round r. The set S is then

Algorithm 8 Winner Beats All – Lazy (WBA-L)

```
Input: set of arms $[n]$, size of choice set $k$, parameter $C$
$t \leftarrow 1$, $r \leftarrow 1$, $A_r \leftarrow [n]$, $a_r \leftarrow \text{Unif}([n])$, $Q \leftarrow \emptyset$
$\widehat{P}_{ij} \leftarrow \frac{1}{2}, \forall i, j \in [n]$
while $t \leq T$ do
    Let $a_t \leftarrow a_r$. Select largest $S \subseteq A_r \setminus \{Q \cup a_t\}$ with $|S| \leq k-1$.
    Play set $S_t \leftarrow S \cup \{a_t\}$ and receive $y_t \in S_t$ as feedback
    $Q \leftarrow Q \cup S$
    For all $i \in S_t$, calculate $\widehat{P}_{ia_t}(t)$ and $\mathcal{J}_i(t, C)$
    if $Q = A_r \setminus \{a_r\}$ then
        $A_{r+1} \leftarrow \emptyset$
        for $i \in [n]$ do
            if $\mathcal{J}_i(t, C) = 0$, then $A_{r+1} \leftarrow A_{r+1} \cup \{i\}$
        end for
        if $\mathcal{J}_{a_r}(t, C) = 0$ then
            $a_{r+1} \leftarrow \text{argmax}_{i \in [n]} \widehat{P}_{ia_r}(t)$
        else
            $a_{r+1} \leftarrow a_r$
        end if
        $Q \leftarrow \emptyset$, $r \leftarrow r + 1$
    end if
    $t \leftarrow t + 1$
end while
```

played with a special arm a_t termed the 'anchor arm'. Both algorithms try to maximize the size of the choice set subject to availability of active arms. In WBA-L the anchor arm has an interpretation of a 'candidate' best arm, whereas in WBA-A the anchor arm is chosen so that one can *quickly* find evidence that arms in S are not good. Hence, in WBA-A an additional requirement on S and a_t is that a_t empirically performs better than each arm in S. Another difference is that WBA-L updates the anchor arm per round, while WBA-A updates it per trial.

Let y_t be the feedback received in trial t when S_t was played including anchor a_t. For all $i, j \in [n]$, let $N_{ij}(t)$ denote the number of times (up to round t) that either arm i or j was chosen when arm j is the anchor, i.e.

$$N_{ij}(t) := \sum_{t'=1}^{t} \mathbb{1}(a_{t'} = j,\, \{i, j\} \subseteq S_{t'},\, y_{t'} \in \{i, j\}) \,. \tag{5.4.1}$$

For each $i, j \in [n]$ and trial t, such that $N_{ij}(t) > 0$, the algorithm maintains an estimate of the *marginal probability* of arm i beating the arm j as

$$\widehat{P}_{ij}(t) := \frac{1}{N_{ij}(t)} \sum_{t'=1}^{t} \mathbb{1}(a_{t'} = j,\ \{i,j\} \subseteq S_{t'},\ y_{t'} = i)\,, \tag{5.4.2}$$

which is the fraction of times i was selected (compared to j) when both i and j were played together and j was the anchor. (When $N_{ij}(t) = 0$, we can simply take $\widehat{P}_{ia}(t)$ to be 1/2.) Similar to Komiyama et al. (2015b), let us define an *empirical divergence* $I_i(t, S)$ which provides a certificate that an arm i is worse than (some) arms in S, as

$$I_i(t,S) = \sum_{j \in S} \mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2})\,,$$

where $d(\widehat{P}_{ij}, \frac{1}{2})$ is the KL-divergence defined as $d(P,Q) = P\log(\frac{P}{Q}) + (1-P)\log(\frac{1-P}{1-Q})$, for $P, Q \in [0,1]$. If $I_i(t,S)$ is 0, it means that arm i is empirically at least as good as all other arms in S, and a higher $I_i(t,S)$ would suggest that arm i is most likely 'bad'. For a constant C, we define the condition $\mathcal{J}_i(t, C)$ for arm $i \in [n]$ and round t as

$$\mathcal{J}_i(t,C) = \mathbb{1}\Big\{\exists S \subseteq [n] : I_i(t,S) \geq |S|\log(nC) + \log(t)\Big\}\,.$$

If $\mathcal{J}_i(t,C) = 1$ for some i, it means that there exists a certificate S to show that i is not likely the best arm as it loses to some arms in S by a large 'margin'.[3] The larger the set S the larger the margin needs to be. This condition can be evaluated in polynomial time by computing $\text{argmax}_{S \subseteq [n]} I_i(t,S) - |S| \cdot \log(nC)$ and checking if it is greater than $\log(t)$. Specifically, we can compute $\text{argmax}_{S \subseteq [n]} I_i(t,S) - |S| \cdot \log(nC)$ by first sorting arms j in the order of values $\mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2})$. We can then start with $S \leftarrow \emptyset$ and add one arm at a time from this sorted ordering to S. We stop adding arms to the set S once the value $\mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2})$ of the current arm j is less than $\log(nC)$. It is easy to see that computing $I_i(t,S) - |S| \cdot \log(nC)$ for this set S gives the value of

[3] Note that the above condition is similar to condition used in Komiyama et al. (2015b), except that they only use the set $[n]$ as a certificate instead of all possible subsets $S \subseteq [n]$. In our analysis and experiments will show that this condition is an improvement over the condition used in Komiyama et al. (2015b) for the case of dueling bandits.

$\text{argmax}_{S \subseteq [n]} I_i(t, S) - |S| \cdot \log(nC)$.

Finally, let t be the final trial in a round r. In order to decide which arms to be included in the next set of active arms A_{r+1} we simply check the condition $\mathcal{J}_i(t, C)$ for each $i \in [n]$ and include all arms for which $\mathcal{J}_i(t, C) = 0$ holds. Note that A_{r+1} can be empty, in which case we will simply play the anchor arm until it set becomes non-empty in the future. The anchor arm in WBA-L is updated for round $r+1$ if $a_r \notin A_{r+1}$, and it becomes the arm that beats a_r with the biggest margin empirically. The anchor arm in each trial in WBA-A is the arm with the best empirical divergence among the set of unplayed arms in that round. Detailed pseudo-code for WBA-L is given in Algorithm 8 and for WBA-A is given in Algorithm 7.

5.5 Regret Bounds

In this section we will provide regret upper bounds for our WBA-A and WBA-L algorithms. The following theorem presents our main result which is a regret bound for our WBA-A algorithm under any choice model belonging to the GCC class.

Theorem 5.5.1 (Regret bound for WBA-A under GCC)**.** *Let n be the number of arms, $k \leq n$ be the choice set size parameter, and i^* be the GCW arm . If the multiway choices are drawn according to a GCC choice model with gap parameters $\{\Delta_{i^*i}\}_{i \neq i^*}$ defined in Equation 5.3.1, and $\Delta_{\min} := \min_{i \neq i^*} \Delta_{i^*i}$, then for any $C \geq 1/\Delta_{\min}^4$, the expected regret incurred by* WBA-A *is upper bounded by*

$$\mathbf{E}\left[R(T)\right] \leq O\left(\frac{n^2 \log n}{\Delta_{\min}^2}\right) + O\left(\sum_{i \in [n] \setminus i^*} \frac{\log(TC)}{\Delta_{i^*i}}\right),$$

where T is the (unknown) time-horizon. Moreover, if the underlying model is MNL with weights $v_1, \cdots, v_n \in \mathbb{R}$, then

$$\mathbf{E}\left[R(T)\right] \leq O\left(\frac{n^2 \log n}{(\Delta_{\min}^{\text{MNL}})^2}\right) + O\left(\sum_{i \in [n] \setminus i^*} \frac{\log(TC)}{\Delta_{i^*i}^{\text{MNL}}}\right).$$

The following theorem gives an upper bound for the WBA-L algorithm under the MNL model.

Theorem 5.5.2 (Regret bound for WBA-L under MNL)**.** *Let n be the number of arms, $k \leq n$ be the choice set size parameter, and $i^* \in [n]$ be the GCW arm. If the multiway choices are drawn according to an MNL model with weights $v_1, \cdots, v_n \in \mathbb{R}$, gap parameters $\Delta_{i^*i}^{\text{MNL}} := \frac{e^{v_{i^*}} - e^{v_i}}{e^{v_{i^*}} + e^{v_i}}$ for $i \in [n]$, and $\Delta_{\min}^{\text{MNL}} := \min_{i \neq i^*} \Delta_{i^*i}^{\text{MNL}}$, then for any $C \geq 1/\left(\Delta_{\min}^{\text{MNL}}\right)^4$, the expected regret incurred by* WBA-L *is upper bounded by*

$$\mathbf{E}\left[R(T)\right] \leq O\left(\sum_{i \in [n] \setminus i^*} \frac{\log(n)\log(TC)}{\Delta_{i^*i}^{\text{MNL}}}\right),$$

where T is the (unknown) time-horizon.

Note that the above upper bound depends on the value of C being larger than $1/\Delta_{\min}^4$ which is an instance-dependent quantity, however, we outline a way to select the parameter C in an instance independent manner.

Remark 1 (Selecting C). A value of T^4 for the parameter C suffices for Theorem 5.5.2 and Theorem 5.5.1 to hold, giving a regret upper bound of $O(\log(TC)) = O(\log(T^5)) = O(\log(T))$. (If T is not known, one can use the doubling trick.) To see this note that in order to obtain any non-trivial upper bound for our algorithm, $\Delta_{\min}$ has to be larger than $1/T$. Hence, either $\Delta_{\min}$ is upper bounded by $1/T$, or the instance is too hard to allow any non-trivial upper bound. Therefore, $C \geq T^4$ would suffice whenever the instance is not already too hard. We actually believe setting $C = T^4$ may be somewhat pessimistic (it arises from taking a union bound over all possible states of the algorithm in our regret analysis (specifically, Lemma 5.7.4) – indeed, in our experiments, we set $C = 1$ for all datasets, and our algorithm still demonstrates sublinear regret with this choice – but it certainly suffices, and the regret bound with $C = T^4$ is at most a constant factor 5 times what one might get with $C = 1$ if the regret bound holds in that case.

Discussion. The above theorems yield an instance-wise $O(n \log n \log T)$ regret bound for the WBA-L algorithm under the MNL model, and an instance-wise $O(n^2 \log n + n \log T)$ regret bound for the WBA-A algorithm under the GCC class of models. Comparing these bounds with the lower bound given in Section 5.3, one can observe the upper bound for WBA-L is *instance-wise optimal* under MNL class, and our bound for the WBA-A algorithm is *asymptotically instance-wise optimal* under GCC. The upper bound for WBA-L is similar to the upper bounds obtained for some early dueling bandit algorithms such as IF (Yue et al., 2009) and BTM (Yue and Joachims, 2011) that make a strong 'linearity' assumption on the arms, while the upper bound for WBA-A is similar to the upper bounds obtained for more recent dueling bandit algorithms such as RUCB (Zoghi et al., 2014) and RMED (Komiyama et al., 2015b) that only assume the existence of a Condorcet winner. It is also important to note that our regret bounds do not depend directly on the choice set size k. However, the behavior of these bound is more subtle and depends on the specific multiway choice model through the gap parameters $\{\Delta_{i^*i}\}_{i \neq i^*}$. We also note that while in general the regret can behave differently for different models, in our experiments, we find that there are choice models (including some in real data) where our algorithms empirically achieve smaller regret when allowed to play sets of size $k > 2$ as compared to $k = 2$. Under the MNL model, the bounds obtained for WBA-L are better than the ones obtained for WBA-A, however, it is important to note that WBA-A is not specialized for MNL and has almost optimal regret for a much larger class of models. Moreover, both our instance-wise bounds under MNL are an improvement over the upper bound for the MaxMinUCB algorithm under MNL for (top-1) choice feedback which depends on worst-case gap parameters (Saha and Gopalan, 2019a).

Proof Overview. Our algorithms are based on the idea of isolating a 'good' anchor arm and playing arms that are competitive against this anchor. Hence, in order to prove a regret upper bound we need to show that the GCW i^* would eventually beat every other arm i, i.e. $\widehat{P}_{i^*i}(t)$ (Equation 5.4.2) would eventually become larger than $1/2$. In this case i^* would become the anchor arm. However, an important technical challenge here is to bound the deviation in these pairwise estimates $\widehat{P}_{i^*i}(t)$ obtained from multiway choices. In the past, Saha and Gopalan (2019b) have shown that if one uses rank breaking to extract pairwise estimates under the MNL model, then these pairwise estimates will be concentrated. However, this concentration result relies crucially on the independence from irrelevant attributes (IIA) property of MNL which states that for any two arms, the odds of choosing one over the other in any set remains the same *regardless of which set is shown.* This concentration result does not apply to our setting as the IIA property does not hold for general GCC models beyond the MNL.

Below we outline a novel coupling argument that allows us to prove concentration for the extracted pairwise estimates between the GCW arm i^* and any other arm $i \in [n]$

Lemma 5.5.1 (Concentration)**.** *Consider a GCC choice model with GCW i^*. Fix $i \in [n]$. Let $S_1, \cdots, S_T$ be a sequence of subsets of $[n]$ and $y_1, \cdots, y_T$ be a sequence of choices according to this model, let $\mathcal{F}_t = \{S_1, y_1, \cdots, S_t, y_t\}$ be a filtration such that S_{t+1} is a measurable function of $\mathcal{F}_t$. We have*

$$\Pr(\widehat{P}_{i^*i}(t) \leq P^{\text{GCC}}_{i^*i} - \epsilon \text{ and } N_{i^*i}(t) \geq m) \leq e^{-d(P^{\text{GCC}}_{i^*i} - \epsilon, P^{\text{GCC}}_{i^*i}) \cdot m} \tag{5.5.1}$$

where

$$P^{\text{GCC}}_{i^*i} = \min_{S:|S|\leq k, \{i^*, i\} \subseteq S} \frac{P_{i^*|S}}{P_{i^*|S} + P_{i|S}}\,, \tag{5.5.2}$$

and $d(\cdot,\cdot)$ is the KL-divergence.

Proof Sketch. Let us consider an alternate process for generating multiway choices y'_t from sets S_t. In this process, given any t and a set S_t such that $i^*, i \in S_t$ with $a_t = i$, we first generate a Bernoulli random variable X_t with probability $P_{i^*|S} + P_{i|S}$. If $X_t = 0$ we set $y'_t = j$ with probability $\frac{P_{j|S}}{1 - P_{i^*|S} - P_{i|S}}$, for $j \in S \setminus \{i, i^*\}$. If $X_t = 1$ then we sample another Bernoulli random variable Z_t with probability $P^{\text{GCC}}_{i^*i}$. If $Z_t = 1$ then we let $y'_t = i^*$, otherwise if $Z_t = 0$ we set $y'_t = i$. Let $P_{i^*i|S_t} = P_{i^*|S_t}/(P_{i^*|S_t} + P_{i|S_t})$. Now, we couple y'_t and y_t as follows: if

$y'_t \in S_t \setminus \{i\}$ then we let $y_t = y'_t$, otherwise if $y'_t = i$ then we let $y_t = i^*$ with probability $(P_{i^*i|S_t} - P^{\mathrm{GCC}}_{i^*i})/(1 - P^{\mathrm{GCC}}_{i^*i})$ and let $y_t = i$ with probability $(1 - P_{i^*i|S_t})/(1 - P^{\mathrm{GCC}}_{i^*i})$. One can verify that y_t is distributed according to the correct underlying choice distribution. It is now easy to observe that the estimates $\widehat{P}_{i^*i}(t)$ under y_t will always be larger than the estimates $\widehat{P}'_{i^*i}(t)$ under y'_t, hence, we will have that $\Pr(\widehat{P}_{i^*i}(t) \leq x) \leq \Pr(\widehat{P}'_{i^*i}(t) \leq x)$ for any $x > 0$. One can then show concentration for the coupled estimates $\widehat{P}'_{i^*i}(t)$, and use it to bound the deviation in $\widehat{P}_{i^*i}(t)$. □

Note that the above lemma only shows concentration for the pairwise estimates $\widehat{P}_{i^*i}(t)$ between i^* and any other arm $i \in [n]$, but not for estimates $\widehat{P}_{ij}(t)$ between two arbitrary arms $i \in [n]$ and $j \in [n]$. However, in order to prove our result we only need concentration of estimates between i^* and any other arm $i \in [n]$. We believe that the above concentration lemma is of independent interest, and might be useful in other learning from multiway choice settings beyond MNL.

Once we have bounded the deviation for the pairwise estimates, we bound the number of rounds r in which i^* is not a part of the active set A_r. We then bound the expected number of times that there exists an arm i such that $\widehat{P}_{i^*i}(t) < \frac{1}{2}$, thus bounding the number of trials until i^* becomes the anchor. Finally, once i^* is the anchor arm, we bound the regret incurred due to sub-optimal arms. We provide detailed proofs of Theorem 5.5.2 and Theorem 5.5.1 in Section 5.7.2.

5.6 Experiments

We compared the performance of our WBA-L and WBA-A algorithms against existing algorithms on our choice bandits problem under different choice models. The first two choice models were MNL models, the next three were from the GCC class, and the last three we choice models extracted from real-world datasets:

1. **MNL-Exp:** A MNL model was generated by drawing random weights from the exponential distribution with parameter $\lambda = 3.5$, i.e. for arm $i \in [n]$, $\log v_i$ as sampled

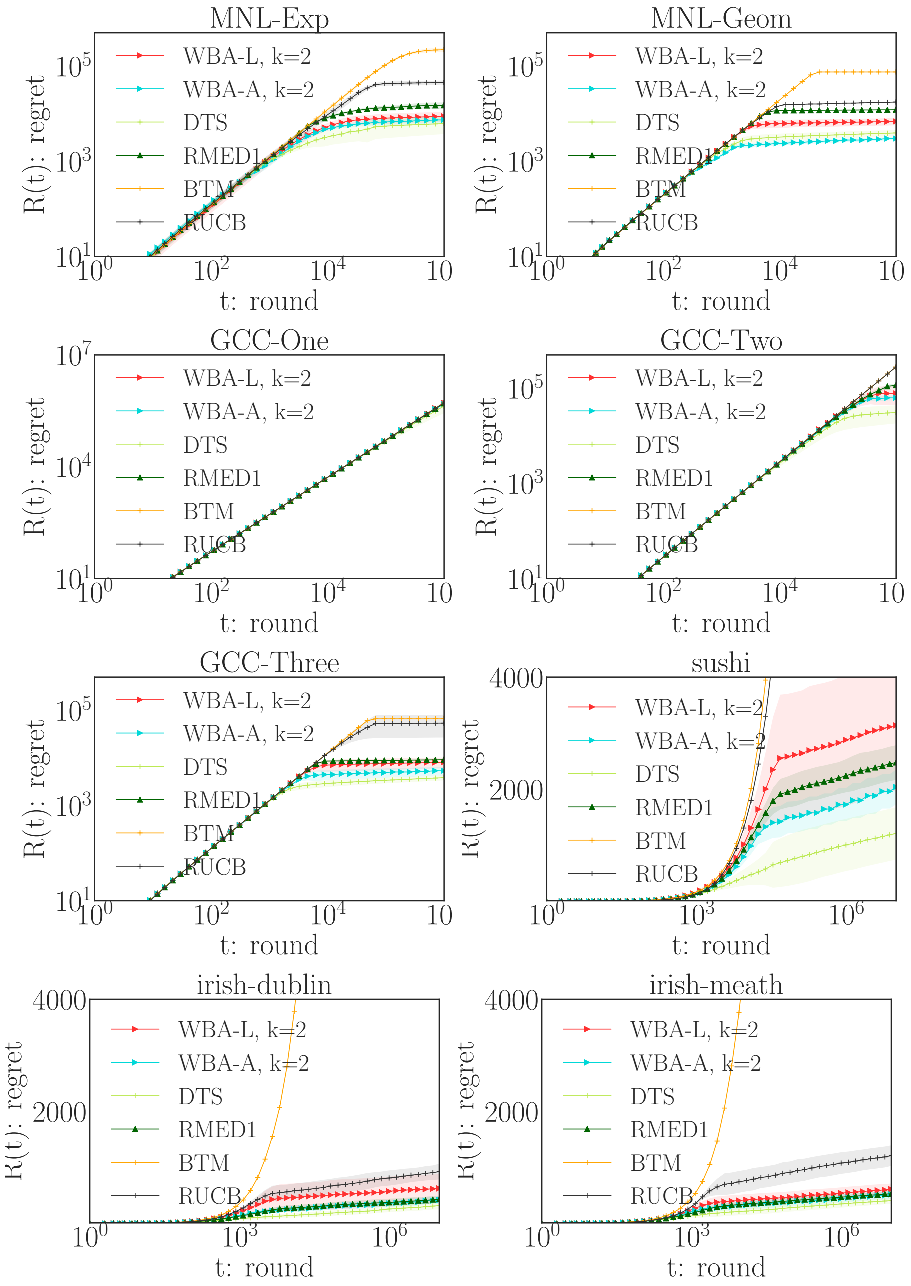


Figure 11: Regret v/s trials for our algorithms WBA-L and WBA-A (for $k = 2$) compared with dueling bandit algorithms (DTS, BTM, RUCB and RMED1) (the shaded region corresponds to std. deviation). As can be observed, our algorithms are competitive against these algorithms.

i.i.d. from $\text{Exp}(\lambda = 3.5)$.

2. **MNL-Geom:** A MNL model was generated with weights $v_1 = e$, $v_2 = e^{\frac{1}{2}}$, …, $v_n = e^{1/2^{n-1}}$.

3. **GCC-One:** This is the choice model from Example 5.2.6, where we selected arm 1 to be the GCW, and for each set S containing arm 1, we set $p_{1|S} = 0.51$ and $p_{i|S} = \frac{0.49}{|S|-1}$ $\forall i \in S \setminus \{1\}$; for sets S not containing the GCW 1, we selected the smallest-index arm in S to be the highest-probability arm i^*_S in S, and set $p_{i^*_S|S} = 0.51$ and $p_{i|S} = \frac{0.49}{|S|-1}$ $\forall i \in S \setminus \{i^*_S\}$).

4. **GCC-Two:** For this choice model, we selected arm 1 to be the GCW, and for each set S we defined $\Delta_S := \min\{\frac{|S|-1}{10}, 0.99\}$. If $i^* \notin S$ we selected the smallest-index arm in S to be the highest-probability arm i^*_S in S, otherwise we let $i^*_S := i^*$. We defined $P_{i^*_S|S} = \frac{1+\Delta_S}{|S|(1-\Delta_S)+2\Delta_S}$ and for any $i \in S \setminus \{i^*_S\}$, $P_{i|S} = \frac{1-\Delta_S}{|S|(1-\Delta_S)+2\Delta_S}$.

5. **GCC-Three:** Here, again, we selected arm 1 to be the GCW, and for each set S we defined $\Delta_S := \max\{\frac{11-|S|}{11}, 0.01\}$. Given this definition of Δ_S, the choice probabilities we defined in a similar manner as GCC-Two.

6. **Sushi:** This is a dataset from (Kamishima, 2003) which contains 5000 partial preference orders given by humans over 100 different types of sushis. Similar to Komiyama et al. (2015a), we selected a subset of 16 sushi types, such that there exists a GCW among them.

7. **Irish-Dublin:** This dataset was also downloaded from *preflib.org* and also contains data about elections held in Dublin, Ireland. The dataset contains $29,988$ partial preference orders given by humans over 9 candidates. We again selected a subset of 8 candidates, such that there exists a GCW among them.

8. **Irish-Meath:** This is a dataset downloaded from *preflib.org* and contains data about elections held in Dublin, Ireland. The dataset contains $64,081$ partial preference orders

given by humans over 14 candidates. We selected a subset of 12 candidates, such that there exists a GCW among them.

Details about extraction of choice model probabilities from real-world datasets can be found in the Appendix. Below we describe the different sets of experiments that were performed. Each experiment was repeated 10 times. The value of n was 100 for all synthetic datasets, 16 for Sushi, 8 for Irish-Dublin, and 12 for Irish-Meath. The parameter C in our algorithms was set to 1.

Comparison with Dueling Bandit Algorithms ($k = 2$). For the special case of $k = 2$, we compared our algorithms with a representative set of dueling bandit algorithms (RMED1 (Komiyama et al., 2015a), DTS (Wu and Liu, 2016), RUCB (Zoghi et al., 2014), BTM (Yue and Joachims, 2011)) for our notion of regret. Note that the purpose of these experiments is merely to perform a sanity check and ensure that our algorithms perform reasonably well compared with dueling bandit baselines when $k = 2$; the goal is not to argue that our choice bandit algorithms beats the state-of-the-art for the specialized dueling bandit ($k = 2$) setting. We set $\alpha = 0.51$ for RUCB and DTS, and $f(K) = 0.3K^{1.01}$ for RMED, and $\gamma = 1.3$ for BTM. Figure 11 contain plots for these comparisons. Our algorithms either perform better or similar to RMED1, RUCB, and BTM on all datasets; and are competitive with DTS on most of the datasets.

Comparison with MaxMinUCB Algorithm (Saha and Gopalan, 2019a) ($k > 2$). We compared the performance of our algorithms with the recent MaxMinUCB algorithm (Saha and Gopalan, 2019a) that was designed and analyzed primarily for MNL choice models under the same notion of regret as ours. [4] We set the parameter α to be 0.51 for MaxMinUCB.

[4] We also considered the SelfSparring algorithm of Sui et al. (2017) and the battling bandit algorithms of Saha and Gopalan (2018), which are applicable to choice models defined in terms of an underlying pairwise comparison model P. However, these algorithms all return *multisets* S_t, and any simple reduction of such multisets to strict sets as considered in our setting (as well as the setting of Saha and Gopalan (2019a)) can end up throwing away important information learned by the algorithms, resulting in a comparison that could

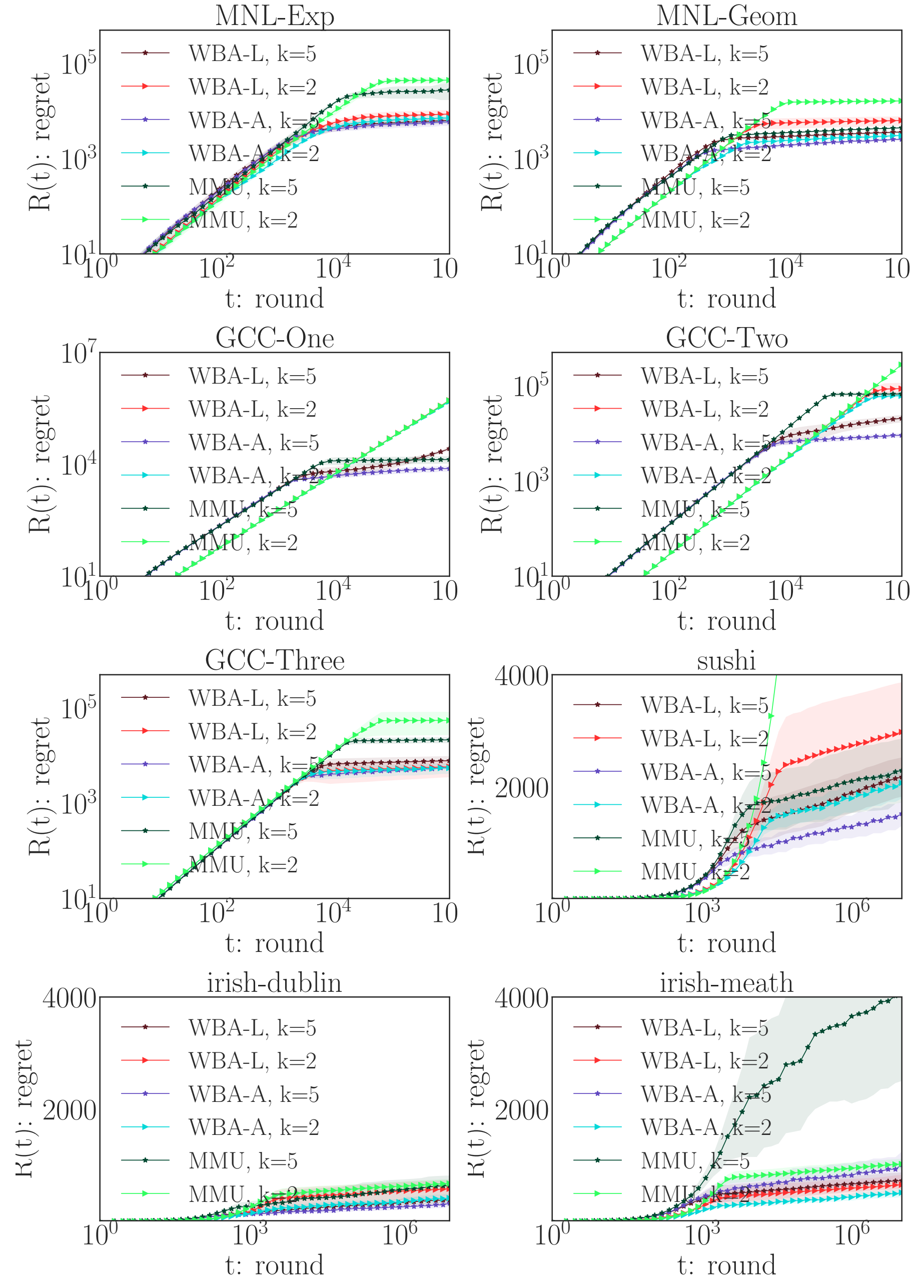


Figure 12: Regret v/s trials for our algorithms WBA-L and WBA-A compared with the MaxMinUCB (MMU) algorithm for $k = 2$ and $k = 5$ (the shaded region corresponds to std. deviation). We observe that our algorithms are better than MaxMinUCB on all datasets for both values of k. We further observe that for several datasets the regret achieved by our algorithm for $k > 2$ is better than the regret of our algorithm for $k = 2$.

Figure 12 contain plots for these experiments for $k = 2$ and $k = 5$. We observe that our algorithms are much better in terms of regret than MaxMinUCB under all datasets for both values of k. One should note that WBA-A performs better than MaxMinUCB even under the MNL datasets, even though MaxMinUCB is specialized to MNL while our algorithms work under more general models. We further observe that under several datasets (GCC-One, GCC-Two, Sushi, Irish-Dublin) the regret achieved by our algorithm for $k > 2$ is better than for $k = 2$. We note that even though our study of more general choice feedback is motivated by applications where it might be desirable to pull sets of size larger than 2 due to reasons other than improving regret, these experimental results show that there exist settings of choice models (including some in real data) where our algorithms empirically achieve a smaller regret when allowed to play sets of size $k > 2$ as compared to $k = 2$.

5.7 Proofs

In this section we provide proofs for the theoretical results in this paper. We will prove the lower bound result given in Section 5.3 and then proceed to the proofs of the regret bounds given in Section 5.5.

5.7.1 Proof of Lower Bound (Theorem 5.3.1)

In order to prove this theorem we will utilize the following change of measure lemma of Kaufmann et al. (2016).

Lemma 5.7.1 (Kaufmann et al. (2016)). *Consider two multi-armed bandit instances where A is the set of arms, and the two different collections of reward distributions are $\boldsymbol{\mu} = \{\mu_i : \forall i \in A\}$ and $\boldsymbol{\mu}' = \{\mu_i' : \forall i \in A\}$, let i_t be the arm played at trial t by an algorithm and X_t be the reward at time t, and let $\mathcal{F}_t = \sigma(i_1, X_1, \cdots, i_t, X_t)$ be the sigma algebra upto time t.*

be unfair to those algorithms. We did explore such reductions and our algorithm easily outperformed them, but we chose not to include the results here due to this issue of fairness. (Moreover, under the MNL model, Saha and Gopalan (2019a) already established that MaxMinUCB outperforms those algorithms – presumably under similar reductions – so in the end, we decided such a comparison would provide little additional value here.)

Consider a $\mathcal{F}_T$ measurable random variable $Z \in [0,1]$, then

$$\sum_{i \in A} \mathbf{E}_{\boldsymbol{\mu}}[N_i(T)] KL(\mu_i, \mu'_i) \geq d(\mathbf{E}_{\boldsymbol{\mu}}[Z], \mathbf{E}_{\boldsymbol{\mu}'}[Z]) \, ,$$

where $N_i(T)$ denotes the number of pulls of arm i in T trials and KL is the Kullback-Leibler divergence between two distributions, and $d(p;q)$ is the Kullback-Leibler divergence between Bernoulli distributions with parameters p and q.

In the proof of the lower bound we first bound the number of times an arm is played using the above lemma, and then bound the total regret due to this arm. Let us first define the regret per arm $i \in [n]$ as

$$R(T,i) = \sum_{t=1}^{T} \mathbb{1}[i \in S_t] \cdot (P_{i^*|S_t \cup i^*} - P_{i|S_t \cup i^*}) \, .$$

We will now provide the proof of the lower bound.

Proof of Theorem 5.3.1. Let us consider an instance $\mathbf{P}$ of the choice bandits problem with n arms such that the best arm i^* is arm 1 and i^* beats all other arms by the largest margin, i.e. $\Delta_{i^*i} \geq \Delta_{ji}$ for any $i,j \in [n]$. Given any set S such that $i \in S$, let i^*_S be the item that has the highest choice probability in S. Note that i^*_S will be equal to i^* when $i^* \in S$. We will assume that for each choice set S there is a unique i^*_S. For any set S and $i \in S$, the instance $\mathbf{P}$ also satisfies that $P_{i^*|S \cup i^*} - P_{i|S \cup i^*} \geq P_{i^*_S|S} - P_{i|S}$. Also, in this instance the ratio of choice probabilities of two different arms in any choice set is bounded by a constant $c > 1$, i.e. $P_{i|S}/P_{j|S} \leq c$ for any $S \subseteq [n]$, $|S| \leq k$, and $i,j \in S$.

For $i \in [n] \setminus \{1\}$, we will now modify this instance to create a new instance $\mathbf{P}'$ where the best arm is i. Now, in the new instance we will have that $P'_{i^*_S|S} := P_{i|S}$ and $P'_{i|S} := P_{i^*_S|S}$ and for all $j \in S \setminus \{i^*_S, i\}$ we will have $P'_{j|S} := P_{j|S}$. Clearly, the best arm in this new instance is the arm i as it has the highest choice probability in any choice set.

Now, given any set S, the probability distributions P_S and P'_S associated with this set are categorical distributions where the feedback is j with probability $P_{j|S}$ and $P_{j'|S}$, respectively. Now, let $A := \{S \subseteq [n] : |S| \leq k\}$ be the set of choice sets of size at most k. We can then use Lemma 5.7.1 with arms corresponding to sets in A and the reward for set S being drawn from categorical distributions P_S and P'_S. We then have the following bound–

$$\sum_{S \in A} \mathbf{E_P}[N_S(T)] KL(P_S, P'_S) \geq d(\mathbf{E_P}[Z], \mathbf{E_{P'}}[Z])\,.$$

where $N_S(T)$ is the number of times set S is played in T rounds, and Z is any $\mathcal{F}_T$ measurable random variable. Also, let $A^i = \{S \in A \setminus \{i\} : i \in S\}$ be all sets that contain i except the singleton set $\{i\}$. Since, we have that for any $S \in A \setminus A^i$ the KL divergence $KL(P_S, P'_S) = 0$, then the above bound becomes:

$$\sum_{S \in A^i} \mathbf{E_P}[N_S(T)] KL(P_S, P'_S) \geq d(\mathbf{E_P}[Z], \mathbf{E_{P'}}[Z])\,.$$

Given any set $S \in A^i$ we can now calculate the KL divergence between the two categorical distributions using the inequality $KL(p, q) \leq \sum_{x \in \mathcal{X}} \frac{(p(x)-q(x))^2}{q(x)}$, where $\mathcal{X}$ is the support of the two distributions.

$$\begin{aligned} KL(P_S, P'_S) &\leq \sum_{j \in S} \frac{(P_{j|S} - P'_{j|S})^2}{P'_{j|S}} \\ &= \frac{(P_{i|S} - P'_{i|S})^2}{P'_{i|S}} + \frac{(P_{i^*_S|S} - P'_{i^*_S|S})^2}{P'_{i^*_S|S}} \\ &= \frac{(P_{i|S} - P_{i^*_S|S})^2}{P_{i^*_S|S}} + \frac{(P_{i|S} - P_{i^*_S|S})^2}{P_{i|S}} \end{aligned}$$

Now, similar to Saha and Gopalan (2019a), let Z be the fraction of times out of T the singleton set $\{i\}$ is played, i.e. $Z = N_i(T)/T$ where $N_i(T)$ counts the number of times set $\{i\}$ is played. We will then have

$$d(\mathbf{E_P}[Z], \mathbf{E_{P'}}[Z]) \geq \left(1 - \frac{\mathbf{E_P}[N_i(T)]}{T}\right) \ln \frac{T}{T - \mathbf{E_{P'}}[N_i(T)]} - \ln 2\,.$$

Since, the algorithm is strongly consistent it can only play a suboptimal arm $\{i\}$ only a sublinear number of times, i.e. $\mathbf{E}_{\mathbf{P}}[N_i(T)] = o(T^\alpha)$ and $T - \mathbf{E}_{\mathbf{P}'}[N_i(T)] = o(T^\alpha)$ for some $\alpha < 1$. Hence, we have that

$$\lim_{T\to\infty} \frac{1}{\ln T} d(\mathbf{E}_{\mathbf{P}}[Z], \mathbf{E}_{\mathbf{P}'}[Z]) \ge \lim_{T\to\infty} \frac{1}{\ln T}\left(1 - \frac{o(T^\alpha)}{T}\right) \ln \frac{T}{o(T^\alpha)} - \ln 2 \ge (1-\alpha)\,. \tag{5.7.1}$$

Combining this with the previous inequality, we have that

$$\lim_{T\to\infty} \frac{1}{\ln T} \sum_{S\in A^i} \mathbf{E}_{\mathbf{P}}[N_S(T)] \left(\frac{(P_{i|S} - P_{i^*_S|S})^2}{P_{i^*_S|S}} + \frac{(P_{i|S} - P_{i^*_S|S})^2}{P_{i|S}} \right) \ge (1-\alpha)\,,$$

which implies

$$\lim_{T\to\infty} \frac{1}{\ln T} \sum_{S\in A^i} \mathbf{E}_{\mathbf{P}}[N_S(T)] \cdot (P_{i|S} - P_{i^*_S|S}) \left(\frac{(P_{i|S} - P_{i^*_S|S})}{P_{i^*_S|S}} + \frac{(P_{i|S} - P_{i^*_S|S})}{P_{i|S}} \right) \ge (1-\alpha)\,,$$

which implies

$$\lim_{T\to\infty} \frac{1}{\ln T} \sum_{S\in A^i} \mathbf{E}_{\mathbf{P}}[N_S(T)] \cdot \frac{3}{2} \cdot (P_{i^*|S\cup i^*} - P_{i|S\cup i^*}) \left(\frac{(P_{i^*_S|S} - P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} - P_{i|S})}{P_{i|S}} \right) \ge (1-\alpha)\,,$$

which follows from the properties of the underlying instance. This implies

$$\lim_{T\to\infty} \frac{1}{\ln T} \mathbf{E}[R(T,i)] \cdot \frac{3}{2} \left(\frac{(P_{i^*_S|S} - P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} - P_{i|S})}{P_{i|S}} \right) \ge (1-\alpha)\,,$$

the last equation follows from the definition of regret per arm. We will now argue that

$$\begin{aligned} \left(\frac{(P_{i^*_S|S} - P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} - P_{i|S})}{P_{i|S}} \right) &\le \Delta_{i^*_S i|S} \cdot \left(\frac{(P_{i^*_S|S} + P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} + P_{i|S})}{P_{i|S}} \right) \\ &\le \Delta_{i^*_S i|S} \cdot (c+3)\,. \end{aligned}$$

Using this we will have that

$$\begin{aligned}
\lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T,\mathcal{A},i)]\left(\frac{(P_{i^*_S|S}-P_{i|S})}{P_{i^*_S|S}}+\frac{(P_{i^*_S|S}-P_{i|S})}{P_{i|S}}\right) &\geq (1-\alpha)\\
\implies \lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T,\mathcal{A},i)]\cdot\Delta_{i^*_S i|S}\cdot(c+3) &\geq (1-\alpha)\\
\implies \lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T,\mathcal{A},i)]\cdot\max_{j\in[n]}\Delta_{ji}\cdot(c+3) &\geq (1-\alpha)\\
\implies \lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T,\mathcal{A},i)]\cdot\Delta_{i^*i}\cdot(c+3) &\geq (1-\alpha)\\
\implies \lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T,\mathcal{A},i)] &\geq \frac{(1-\alpha)}{c+3}\cdot\frac{1}{\Delta_{i^*i}}
\end{aligned}$$

where $\Delta_{ji} := \max_{S:|S|\leq k,\{j,i\}\subseteq S}\frac{P_{j|S}-P_{i|S}}{P_{j|S}+P_{i|S}}$ and the second last inequality holds because of the property of the underlying instance. Since, we have that $R(T)=\sum_{i\in[n]}R(T,i)$ we get that

$$\lim_{T\to\infty}\frac{1}{\ln T}\mathbf{E}[R(T)]\geq\Omega\left(\sum_{i\neq i^*}\frac{1}{\Delta_{i^*i}}\right),$$

which concludes the proof of the lower bound for the general GCC class.

Now, given any MNL instance, we also derive a regret lower bound which gives the minimum instance-wise regret any strongly-consistent algorithm for the GCC class needs to incur under this MNL instance.

Consider an instance $\mathbf{P}$ with an underlying MNL model with weights $v_1,\cdots,v_n$. We will assume that all these weights are distinct for simplicity, otherwise we can add a small perturbation to these weights to break ties. We will re-parameterize this instance, and let $w_i := \log v_i$ for any $i\in[n]$. Given any set S, let $w_S=\sum_{j\in[n]}w_j$. We have that $P_{i|S}=w_i/w_S$ for any $i\in S$. Given S, we will again let i^*_S to be the arm that has the highest choice probability in S, i.e. $i^*_S=\text{argmax}_{i\in S}w_i$. We will denote by κ the ratio of the maximum weight to minimum weight, i.e. $\kappa=\max_i w_i/\min_j w_j$.

For $i\in[n]\setminus\{1\}$, we will now modify this instance to create a new instance $\mathbf{P'}$ where the GCW arm is i. In the new instance, for any set S, we will have that $P'_{i^*_S|S} := P_{i|S}$ and $P'_{i|S} := P_{i^*_S|S}$ and for all $j \in S \setminus \{i^*_S, i\}$ we will have $P'_{j|S} := P_{j|S}$. Clearly, the best arm in this new instance is the arm i as it has the highest choice probability in any choice set. It is also easy to verify that this new instance $\mathbf{P}'$ belongs to the GCC class. Note that $\mathbf{P}'$ might not belong to the MNL class. Under the instance $\mathbf{P}$ we have that $(1+\kappa)(P_{i^*|S \cup i^*} - P_{i|S \cup i^*}) \geq (P_{i^*_S|S} - P_{i|S})$.

Given these two instances, we can follow steps analogous to the proof of the GCC case, to derive the following bound

$$\lim_{T \to \infty} \frac{1}{\ln T} \mathbf{E}[R(T,i)] \cdot (1+\kappa) \left(\frac{(P_{i^*_S|S} - P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} - P_{i|S})}{P_{i|S}} \right) \geq (1-\alpha)\,.$$

We now have that

$$\begin{aligned} \left(\frac{(P_{i^*_S|S} - P_{i|S})}{P_{i^*_S|S}} + \frac{(P_{i^*_S|S} - P_{i|S})}{P_{i|S}} \right) &= \frac{w_{i^*_S} - w_i}{w_i} + \frac{w_{i^*_S} - w_i}{w_{i^*_S}} = \frac{w_{i^*_S} - w_i}{w_{i^*_S} + w_i} \left(\frac{w_{i^*_S} + w_i}{w_i} + \frac{w_{i^*_S} + w_i}{w_{i^*_S}} \right) \\ &\leq \frac{w_{i^*} - w_i}{w_{i^*} + w_i} (3+\kappa) = \Delta^{\text{MNL}}_{i^* i} (3+\kappa) \end{aligned}$$

Using the same steps as above we have that

$$\lim_{T \to \infty} \frac{1}{\ln T} \mathbf{E}[R(T,i)] \geq (1-\alpha) \cdot \frac{1}{\Delta^{\text{MNL}}_{i^* i}} \cdot \frac{1}{(3+\kappa)(1+\kappa)}\,.$$

Since, we have that $R(T) = \sum_{i \in [n]} R(T,i)$ we get that

$$\lim_{T \to \infty} \frac{1}{\ln T} \mathbf{E}[R(T)] = \Omega \left(\sum_{i \in [n] \setminus \{i^*\}} \frac{1}{\Delta^{\text{MNL}}_{i^* i}} \right),$$

which concludes the proof of the lower bound for the MNL case.

□

Note that the lower bound for the MNL model also implies a lower bound for the general GCC class. However, we chose to construct an instance outside MNL for the GCC lower

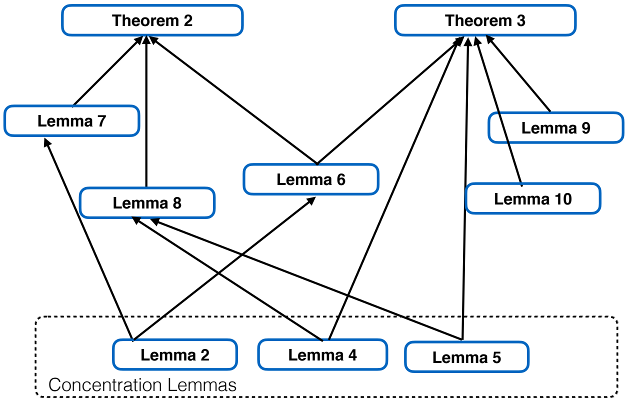


Figure 13: A flow-chart giving organization for the proof of Theorem 5.5.2 and Theorem 5.5.1

bound in order to show that such a lower bound also holds beyond the MNL. Also, note that the lower bound in Saha and Gopalan (2019a) for MNL under MNL consistent algorithms is worst-case while our lower bound for MNL under GCC consistent algorithms applies to all MNL instances.

5.7.2 Proof of Upper Bound Results

In this section we will present proofs for our upper bound results. We will first define some additional notation in Section 5.7.2.1. In Section 5.7.2.2 we prove concentration results that will be useful in proving our regret bounds. We will then proceed to the proofs of regret bounds in Section 5.7.2.3 and Section 5.7.2.4. Figure 13 gives an overview of the proof structure.

5.7.2.1 Additional Notation

Let $M_{ij}(t)$ be the number of times i is played when j is the anchor up to trial t, i.e.

$$M_{ij}(t) := \sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i, j\} \subseteq S_{t'}] \,. \tag{5.7.2}$$

Given a trial t and arms $i, j \in [n]$, let $P^t_{i|ij}$ be the choice probability of i in sets S_t where j is the anchor arm, averaged across t rounds, i.e.

$$P^t_{i|ij} := \frac{\sum_{t'=1}^{t} P_{i|S_{t'}} \cdot \mathbb{1}[a_{t'} = j, \{i, j\} \subseteq S_{t'}]}{M_{ij}(t)}, \tag{5.7.3}$$

and let $\widehat{P}^t_{i|ij}$ be an empirical estimate of $P^t_{i|ij}$, i.e.

$$\widehat{P}^t_{i|ij} := \frac{\sum_{t'=1}^{t} \mathbb{1}[y_{t'} = i] \cdot \mathbb{1}[a_{t'} = j, \{i, j\} \subseteq S_{t'}]}{M_{ij}(t)}. \tag{5.7.4}$$

We will also define the following time-dependent gap quantity.

$$\Delta^t_{i^*i} := \frac{P^t_{i^*|ii^*} - P^t_{i|ii^*}}{P^t_{i^*|ii^*} + P^t_{i|ii^*}}.$$

Let us define the regret per arm $i \in [n]$ for a set S as

$$r(S, i) = \mathbb{1}[i \in S] \cdot (P_{i^*|S\cup i^*} - P_{i|S\cup i^*}). \tag{5.7.5}$$

Finally, let us also denote by R^t_{ij} the regret up to time t incurred during times when arm j is the anchor arm, i.e.

$$R_{ij}(t) := \sum_{t'=1}^{t} r(S_{t'}, i) \cdot \mathbb{1}[a_{t'} = j, \{i, j\} \subseteq S_{t'}]. \tag{5.7.6}$$

where $r(S_{t'}, i)$ is the instantaneous regret for arm i at time t' defined in Equation 5.7.5. Note that

$$R_{ii^*}(t) = M_{i^*i}(t) \cdot (P^t_{i^*|ii^*} - P^t_{i|ii^*}).$$

We will also define $R_{i|i^*}(t)$ as the regret for arm i when arm i is the anchor and i^* is also played together with it, i.e.

$$R_{i|i^*}(t) := \sum_{t'=1}^{t} r(S_{t'}, i) \cdot \mathbb{1}[a_{t'} = i, \{i, i^*\} \subseteq S_{t'}]. \tag{5.7.7}$$

Note that

$$R_{i|i^*}(t) = M_{i^*i}(t) \cdot (P^t_{i^*|i^*i} - P^t_{i|i^*i})\,.$$

5.7.2.2 Concentration Inequalities

In this section we will prove all the concentration inequalities required to prove our regret upper bounds. These concentration inequalities are needed to bound the deviation in the pairwise preference estimates extracted from multiway comparisons.

Lemma 5.5.1. Consider a GCC choice model with GCW i^*. Fix $i \in [n]$. Let $S_1, \cdots, S_T$ be a sequence of subsets of $[n]$ and $y_1, \cdots, y_T$ be a sequence of choices according to this model, let $\mathcal{F}_t = \{S_1, y_1, \cdots, S_t, y_t\}$ be a filtration containing the history of execution of the algorithm such that S_{t+1} is a measurable function of $\mathcal{F}_t$. Let $\widehat{P}_{i^*i}(t)$ be the empirical probability estimate of i^* beating i calculated according to Equation 5.4.2, then for any given $t \in [T]$ we have that

$$\Pr(\widehat{P}_{i^*i}(t) \le P^{\text{GCC}}_{i^*i} - \epsilon \text{ and } N_{i^*i}(t) \ge m) \le e^{-d(P^{\text{GCC}}_{i^*i} - \epsilon, P^{\text{GCC}}_{i^*i}) \cdot m} \tag{5.7.8}$$

where

$$P^{\text{GCC}}_{i^*i} = \min_{S:|S| \le k, \{i^*, i\} \subseteq S} \frac{P_{i^*|S}}{P_{i^*|S} + P_{i|S}}\,, \tag{5.7.9}$$

and $d(\cdot,\cdot)$ is the KL-divergence between two Bernoulli distributions, and $N_{i^*i}(t) := \sum_{t'=1}^{t} \mathbb{1}(a_{t'} = i,\, \{i^*, i\} \subseteq S_{t'},\, y_{t'} \in \{i^*, i\})$. The above bound implies the following bound

$$\Pr(\widehat{P}_{i^*i}(t) \le \frac{1}{2}; N_{i^*i}(t) \ge m) \le e^{-d(\frac{1}{2}, P^{\text{GCC}}_{i^*i})m} \tag{5.7.10}$$

We also have the following bound–

$$\Pr(\widehat{P}_{ii^*}(t) \ge P^{\text{GCC}}_{ii^*} + \epsilon; N_{i^*i}(t) \ge m) \le e^{-d(P^{\text{GCC}}_{i^*i} - \epsilon, P^{\text{GCC}}_{i^*i}) \cdot m} \tag{5.7.11}$$

where $P^{\text{GCC}}_{ii^*} = 1 - P^{\text{GCC}}_{i^*i}$.

Proof. We will first prove inequality 5.7.8. Let $Z_1, Z_2, \cdots$ be a sequence of i.i.d. Bernoulli random variables with probability of success $P^{\text{GCC}}_{i^*i}$. We will initialize a counter C to 0. Let us consider an alternate process for generating multiway choices y'_t from sets S_t. In this process, given any t and a set S_t such that $i^*, i \in S_t$ with $a_t = i$, we first generate a Bernoulli random variable X_t with probability $P_{i^*|S} + P_{i|S}$. If $X_t = 0$ we sample a multinomial random variable Y_t such that $Y_t = j$ with probability $\frac{P_{j|S}}{1-P_{i^*|S}-P_{i|S}}$, for $j \in S \setminus \{i, i^*\}$, and let $y'_t = Y_t$. If $X_t = 1$, then we increase the counter C by 1, and sample the Bernoulli random variable Z_C with probability $P^{\text{GCC}}_{i^*i}$. If $Z_C = 1$ we declare i^* as the choice, i.e. $y'_t = i^*$, otherwise if $Z_C = 0$ we declare i to be the choice. Let $P_{i^*i|S} = P_{i^*|S}/(P_{i^*|S} + P_{i|S})$. Now, we couple the process generating y'_t and the process generating y_t as follows: if $y'_t \in S_t \setminus \{i\}$ then we let $y_t = y'_t$, otherwise if $y'_t = i$ then we let $y_t = i^*$ with probability $(P_{i^*i|S_t} - P^{\text{GCC}}_{i^*i})/(1 - P^{\text{GCC}}_{i^*i})$ and let $y_t = i$ with probability $(1 - P_{i^*i|S_t})/(1 - P^{\text{GCC}}_{i^*i})$. The first thing to check is that y_t is drawn from the correct probabilities $P_{y_t|S_t}$ according to the underlying choice model. We have, for any $j \in S_t \setminus \{i^*, i\}$

$$\begin{aligned}
\Pr y_t = j|S_t &= \Pr X_t = 0, Y_t = j|S_t \\
&= \Pr X_t = 0|S_t \Pr Y_t = j|X_t = 0, S_t \\
&= \left(1 - P_{i^*|S_t} - P_{i|S_t}\right) \cdot \frac{P_{j|S_t}}{1 - P_{i^*|S_t} - P_{i|S_t}} \\
&= P_{j|S_t}
\end{aligned}$$

We also have that

$$\begin{aligned}
\Pr y_t = i^*|S_t &= \Pr X_t = 1, Y_t = i^*|S_t + \frac{P_{i^*i|S_t} - P^{\text{GCC}}_{i^*i}}{1 - P^{\text{GCC}}_{i^*i}} \cdot \Pr X_t = 1, Y_t = i|S_t \\
&= \left(P_{i^*|S_t} + P_{i|S_t}\right) \cdot \left(P^{\text{GCC}}_{i^*i} + (1 - P^{\text{GCC}}_{*i}) \cdot \frac{P_{i^*i|S_t} - P^{\text{GCC}}_{i^*i}}{1 - P^{\text{GCC}}_{i^*i}}\right) \\
&= \left(P_{i^*|S_t} + P_{i|S_t}\right) \cdot \left(P_{i^*i|S_t}\right) \\
&= P_{i^*|S_t}
\end{aligned}$$

where the last inequality follows from definition of $P_{i^*i|S}$. The fact that $\Pr y_t = i|S_t = P_{i|S}$ follows from the fact that the choice probabilities sum to 1.

Let $W_{i^*i}(t) = \sum_{t'=1}^{t} \mathbb{1}(a_{t'} = i,\, \{i^*, i\} \subseteq S_{t'},\, y_{t'} = i^*)$ and $W'_{i^*i}(t) = \sum_{t'=1}^{t} \mathbb{1}(a_{t'} = i,\, \{i^*, i\} \subseteq S_{t'},\, y'_{t'} = i^*)$. Due to the above coupling, we immediately have that $\Pr(W_{i^*i}(t)) \geq \Pr(W'_{i^*i}(t))$ for any $t \in [T]$. Then

$$\Pr(W_{i^*i}(t) \leq r) \leq \Pr(W'_{i^*i}(t) \leq r)$$

for any $r \geq 0$, and any $t \in [T]$. Using this, we have that

$$\begin{aligned}
\Pr(\widehat{P}_{i^*i}(t) \leq P^{\text{GCC}}_{i^*i} - \epsilon; N_{i^*i}(t) \geq m) &= \Pr(W_{i^*i}(t) \leq N_{i^*i}(t) \cdot (P^{\text{GCC}}_{i^*i} - \epsilon); N_{i^*i}(t) \geq m) \\
&\leq \Pr(W'_{i^*i}(t) \leq N_{i^*i}(t) \cdot (P^{\text{GCC}}_{i^*i} - \epsilon); N_{i^*i}(t) \geq m)
\end{aligned}$$

Now, using techniques similar to Saha and Gopalan (2019b), we have the following bound

$$\begin{aligned}
\Pr(\frac{W'_{i^*i}(t)}{N_{i^*i}(t)} \leq P^{\text{GCC}}_{i^*i} - \epsilon; N_{i^*i}(t) \geq m) &= \Pr(\frac{\sum_{s=1}^{N_{i^*i}(t)} Z_s}{N_{i^*i}(t)} \leq P^{\text{GCC}}_{i^*i} - \epsilon; N_{i^*i}(t) \geq m) \\
&= \sum_{r=m}^{t} \Pr(\frac{\sum_{s=1}^{r} Z_s}{r} \leq P^{\text{GCC}}_{i^*i} - \epsilon; N_{i^*i}(t) = r) \\
&= \sum_{r=m}^{t} \Pr(\frac{\sum_{s=1}^{r} Z_s}{r} \leq P^{\text{GCC}}_{i^*i} - \epsilon) \Pr(N_{i^*i}(t) = r)
\end{aligned}$$

where the last equality holds because of the fact that $Z_1, Z_2, \cdots$ is an independent sequence of random variables that do not lie in the sigma algebra of $S_1, \cdots, S_t, X_1, \cdots, X_t$. Using the KL-divergence based concentration inequality from Garivier and Cappé (2011) we have that

$$\Pr(\frac{\sum_{s=1}^{r} Z_s}{r} \leq P_{i^*i}^{\mathrm{GCC}} - \epsilon) \leq e^{-d(P_{i^*i}^{\mathrm{GCC}} - \epsilon, P_{i^*i}^{\mathrm{GCC}})r} .$$

We then have that

$$\begin{aligned} \sum_{r=m}^{t} \Pr(\frac{\sum_{s=1}^{r} Z_s}{r} \leq P_{i^*i}^{\mathrm{GCC}} - \epsilon) \Pr(N_{i^*i}(t) = r) &\leq \sum_{r=m}^{t} e^{d(P_{i^*i}^{\mathrm{GCC}} - \epsilon, P_{i^*i}^{\mathrm{GCC}})r} \Pr(N_{i^*i}(t) = r) \\ &\leq e^{-d(P_{i^*i}^{\mathrm{GCC}} - \epsilon, P_{i^*i}^{\mathrm{GCC}})m} \end{aligned}$$

The proof of reverse direction follows from a similar coupling argument followed by the above concentration inequality. □

Note that the above coupling technique has similarity to the coupling used in Saha and Gopalan (2019b) in order to show concentration of pairwise estimates under the MNL model. However, this argument relies on the IIA property of MNL, which does not hold under general GCC models.

The above concentration inequality is, however, not enough to prove a tight instance-wise bound for WBA-L and WBA-A, as it bounds the worst case probabilities $P_{i^*i}^{\mathrm{GCC}}$. In order to achieve a tight instance-wise bound we will develop new instance-wise concentration inequalities using martingale based argument. This new concentration bound is a contribution of this paper and was not present in the conference version (Agarwal et al., 2020).

Lemma 5.7.2. *Let $S_1, \cdots, S_T$ be a sequence of subsets of $[n]$ and $y_1, \cdots, y_T$ be a sequence of choices according to this model, let $\mathcal{F}_t = \{S_1, y_1, \cdots, S_t, y_t\}$ be a filtration such that S_{t+1}*

is a measurable function of $\mathcal{F}_t$. Given $\lambda > 0$, for any $t \in [T]$ and any $i \in [n]$, we have that

$$\Pr(\left|\frac{\widehat{P}^t_{i|ii^*}}{\widehat{P}^t_{i^*|ii^*} + \widehat{P}^t_{i|ii^*}} - \frac{P^t_{i|ii^*}}{P^t_{i^*|ii^*} + P^t_{i|ii^*}}\right| \geq \sqrt{\frac{2\Delta^t_{i^*i}\lambda}{R_{ii^*}(t)}} + \frac{2\Delta^t_{i^*i}\lambda}{3R_{ii^*}(t)}) \leq 4nT\log(T) \cdot e^{-\lambda} + 8nT \cdot e^{-\lambda/4}\,, \tag{5.7.12}$$

where the quantities $\widehat{P}^t_{i|ii^*}$, $\widehat{P}^t_{i^*|ii^*}$, $\Delta^t_{i^*i}$ and $R_{ii^*}(t)$ are defined in Section 5.7.2.1. Moreover, if the underlying model is MNL, then for any $i \in [n]$ and $t \in [T]$, we have that

$$\Pr(\left|\frac{\widehat{P}^t_{i|i^*i}}{\widehat{P}^t_{i^*|i^*i} + \widehat{P}^t_{i|i^*i}} - \frac{w_i}{w_i + w_{i^*}}\right| \geq \sqrt{\frac{2\Delta^{\mathrm{MNL}}_{i^*i}\lambda}{R_{i|i^*}(t)}} + \frac{2\Delta^{\mathrm{MNL}}_{i^*i}\lambda}{3R_{i|i^*}(t)}) \leq 4nT\log(T) \cdot e^{-\lambda} + 8nT \cdot e^{-\lambda/4}\,, \tag{5.7.13}$$

where $R_{i|i^*}(t)$ is defined in Section 5.7.2.1. Moreover, if the underlying model is MNL, for any $i, j \in [n]$ with $w_{i^*} - w_j \leq w_j - w_i$ and any $t \in [T]$ we have that

$$\Pr(\left|\frac{\widehat{P}^t_{i|ij}}{\widehat{P}^t_{j|ij} + \widehat{P}^t_{i|ij}} - \frac{w_i}{w_i + w_j}\right| \leq \sqrt{\frac{4\Delta^{\mathrm{MNL}}_{ji}\lambda}{R_{ij}(t)}} + \frac{4\Delta^{\mathrm{MNL}}_{ji}\lambda}{3R_{ij}(t)}) \leq 4nT\log(T) \cdot e^{-\lambda} + 8nT \cdot e^{-\lambda/4}\,, \tag{5.7.14}$$

where all the quantities are again defined in Section 5.7.2.1.

Proof. Fix an arm i and anchor arm j. In order to prove this lemma we will first bound the deviation between $\widehat{P}^t_{i|ij}$ and $P^t_{i|ij}$. In order to bound this deviation we define X_t to be an indicator random variable denoting the event that arm i won in trial t when j was the anchor, i.e. $X_t := \mathbb{1}[y_t = i, a_t = j, \{i, j\} \subseteq S_t]$. Also, let Y_t be an indicator random variable denoting the event that i was played in trial t and j was the anchor arm, i.e. $Y_t := \mathbb{1}[a_t = j, \{i, j\} \subseteq S_t]$. Note that $R_{ij}(t) = \sum_{t'=1}^{t} r(S_{t'}, i) \cdot Y_{t'}$. We also define Z_t as follows:

$$Z_t := \sum_{t'=1}^{t} \left(X_{t'} - P_{i|S_{t'}} \cdot Y_{t'}\right).$$

We can then write the deviation between $\widehat{P}^t_{i|ij}$ and $P^t_{i|ij}$ in terms of the random variable Z_t

as follows:

$$\Pr\left(\left|\widehat{P}^t_{i|ij} - P^t_{i|ij}\right| \geq \epsilon\right) \leq \Pr\left(\left|\sum_{t'=1}^{t}\left(X_{t'} - P_{i|S_{t'}} \cdot Y_{t'}\right)\right| > 2M_{ij}(t)\epsilon\right) = \Pr\left(\left|Z_t\right| > M_{ij}(t)\epsilon\right), \tag{5.7.15}$$

for any $\epsilon > 0$. We will show that the random variables $\{Z_t\}$ form a martingale sequence with respect to the filtration $\mathcal{F}_{t-1}$. To see this we will calculate $\mathbf{E}[Z_t|\mathcal{F}_{t-1}]$ as follows

$$\begin{aligned}\mathbf{E}[Z_t|\mathcal{F}_{t-1}] &= \mathbf{E}[Z_{t-1}|\mathcal{F}_{t-1}] + \mathbf{E}\Big[X_t - P_{i|S_t} \cdot Y_t\Big|\mathcal{F}_{t-1}\Big] \\ &= Z_{t-1} + \mathbf{E}\Big[X_t\Big|\mathcal{F}_{t-1}\Big] - P_{i|S_t} \cdot Y_t\,.\end{aligned}$$

The second equality holds because Y_t is a deterministic quantity given $\mathcal{F}_{t-1}$. In the case $Y_t = 0$ we have that $X_t = 0$; in the case that $Y_t = 1$, X_t is a Bernoulli random variable with probability $P_{i|S_t}$. Hence, in both cases we have that $\mathbf{E}[X_t|\mathcal{F}_{t-1}] - P_{i|S_t} \cdot Y_t = 0$. This implies that

$$\mathbf{E}[Z_t|\mathcal{F}_{t-1}] = Z_{t-1}\,.$$

Hence, we have shown that the sequence Z_t's form a martingale sequence. We can now use the Bernstein inequality for martingales (Cesa-Bianchi and Lugosi, 2006) (See Appendix) to bound the probability in Equation 5.7.15. This inequality bounds the deviation in Z_t using information about the second moments of the sequence. Let

$$\sigma_t^2 := \sum_{t'=1}^{t} \mathbf{E}\Big[(X_{t'} - P_{i|S_{t'}} \cdot Y_{t'})^2|\mathcal{F}_{t'-1}\Big]\,.$$

We now calculate the value of σ_t^2. Recall that if $Y_t = 0$ then $X_t = 0$; and if $Y_t = 1$ then X_t is a Bernoulli random variable with probability $P_{i|S_t}$. We then have that

$$\begin{aligned}\sigma_t^2 &= \sum_{t'=1}^{t} \mathbf{E}\Big[(X_{t'} - P_{i|S_t} \cdot Y_{t'})^2 | \mathcal{F}_{t'-1}\Big] \\ &= \sum_{t'=1}^{t} Y_{t'} \cdot \mathrm{Var}(X_{t'} | \mathcal{F}_{t'-1}, Y_{t'} = 1) + (1 - Y_{t'}) \cdot 0 \\ &= \sum_{t'=1}^{t} Y_{t'} \cdot P_{i|S_t}(1 - P_{i|S_t}) \leq M_{ij}(t) \cdot P_{i|ij}^t \,. \end{aligned} \tag{5.7.16}$$

We then have

$$\Pr(|Z_t| \geq M_{ij}(t)\epsilon) \leq \Pr(|Z_t| \geq M_{ij}(t)\epsilon, \sigma_t^2 \leq M_{ij}(t) \cdot P_{i|ij}^t) \,,$$

for any $\epsilon > 0$. Also, $|X_t - P_{i|S_t} \cdot Y_t| \leq 1$. Using the Bernstein's inequality for martingales, we have that,

$$\Pr\Big(|Z_t| > \sqrt{2\nu\lambda} + 2\lambda/3, \sigma_t^2 \leq \nu\Big) \leq 2e^{-\lambda} \,,$$

for any constants $\lambda, \nu > 0$. However, the problem with our desired bound is that we want to bound the deviation of Z_t by a quantity that depends on $P_{i|ij}^t$ and $M_{ij}(t)$ which are random variables, whereas in the above inequality we need λ and ν to be constants. We use the peeling technique (Bartlett et al. (2005)), and break down the process into different variance classes. We will then take a union bound over all the variance classes, i.e. values of $M_{ij}(t) \cdot P_{i|ij}^t$.

Let us define $f(\nu, \lambda) = \sqrt{2\nu\lambda} + 2\lambda/3$ for any ν, λ. We then have

$$\begin{aligned}
&\Pr\Big(|Z_t| > f(M_{ij}(t)P^t_{i|ij}, \lambda)\,,\, \sigma_t^2 \le M_{ij}(t)P^t_{i|ij}\Big) \\
&\quad\le \sum_{r=1}^{\lceil \log(t) \rceil} \Pr\Big(\frac{t}{2^r} < M_{ij}(t)P^t_{i|ij} \le \frac{t}{2^{r-1}}\,,\, |Z_t| > f(\alpha M_{ij}(t), \lambda)\,,\, \sigma_t^2 \le M_{ij}(t)P^t_{i|ij}\Big) \\
&\qquad\qquad + \Pr\Big(0 \le M_{ij}(t)P^t_{i|ij} \le 1\,,\, |Z_t| > f(M_{ij}(t)P^t_{i|ij}, \lambda)\,,\, \sigma_t^2 \le \alpha M_{ij}(t)\Big) \\
&\quad\le \sum_{r=1}^{\lceil \log(t) \rceil} \Pr\Big(|Z_t| > f(\frac{t}{2^r}, \lambda)\,,\, \sigma_t^2 \le \frac{t}{2^{r-1}}\Big) \\
&\qquad\qquad + \Pr\Big(|Z_t| > f(0, \lambda)\,,\, \sigma_t^2 \le 1\Big)\,.
\end{aligned}$$

The last two inequalities are due to the union bound. We now use the Bernstein's inequality to bound the above as:

$$\begin{aligned}
\Pr\Big(|Z_t| > f(\gamma, \lambda)\,, \sigma_t^2 \le 2\gamma\Big) &\le \Pr\Big(|Z_t| > \sqrt{4\gamma\lambda} + 2\lambda/3\,, \sigma_t^2 \le 2\gamma\Big) \\
&\le 2e^{-\lambda}\,.
\end{aligned}$$

We also have that, for any $\lambda \ge 1$,

$$\begin{aligned}
\Pr\Big(|Z_t| > f(0, \lambda)\,, \sigma_t^2 \le 1\Big) &\le \Pr\Big(|Z_t| > \lambda\,, \sigma_t^2 \le 1\Big) \\
&\le 2e^{-\frac{\lambda^2}{2(1+2\lambda/3)}} \\
&\le 2e^{-\frac{\lambda^2}{2(\lambda+2\lambda/3)}} \\
&\le 2e^{-\frac{\lambda^2}{4\lambda}} \le e^{-\lambda/4}\,.
\end{aligned}$$

Combining this all together, we have that

$$\Pr\Big(|Z_t| > \sqrt{2M_{ij}(t)P^t_{i|ij}\lambda} + 2\lambda/3\Big) \le 2\log(t)e^{\lambda} + 4e^{-\lambda/4}\,.$$

Using this, we have that

$$\Pr\left(\left|\widehat{P}^t_{i|ij} - P^t_{i|ij}\right| \geq \sqrt{\frac{2P^t_{i|ij}\lambda}{M_{ij}(t)}} + \frac{2\lambda}{3M_{ij}(t)}\right) \leq 2\log(t)e^{\lambda} + 4e^{-\lambda/4}\,. \tag{5.7.17}$$

Using the same argument as above we can also show that

$$\Pr\left(\left|\widehat{P}^t_{j|ij} - P^t_{j|ij}\right| \geq \sqrt{\frac{2P^t_{j|ij}\lambda}{M_{ij}(t)}} + \frac{2\lambda}{3M_{ij}(t)}\right) \leq 2\log(t)e^{\lambda} + 4e^{-\lambda/4}\,.$$

Using the above we have that

$$\begin{aligned}\Pr(\left|\frac{\widehat{P}^t_{i|ij}}{\widehat{P}^t_{j|ij} + \widehat{P}^t_{i|ij}} - \frac{P^t_{i|ij}}{P^t_{j|ij} + P^t_{i|ij}}\right| \geq \sqrt{\frac{2\lambda}{M_{ij}(t)\cdot(P^t_{j|ij} + P^t_{i|ij})}} + \frac{2\lambda}{3M_{ij}(t)\cdot(P^t_{j|ij} + P^t_{i|ij})}) \\ \leq 4\log(t)e^{\lambda} + 8e^{-\lambda/4}\,.\end{aligned} \tag{5.7.18}$$

We will prove the first part of the lemma (Equation 5.7.12) where i^* is the anchor arm, using the fact that $R_{ii^*}(t) = M_{ii^*}(t)\cdot(P^t_{i^*|ii^*} - P^t_{i|ii^*})$ to get that

$$\begin{aligned}\Pr(\left|\frac{\widehat{P}^t_{i|ii^*}}{\widehat{P}^t_{i^*|ii^*} + \widehat{P}^t_{i|ii^*}} - \frac{P^t_{i|ii^*}}{P^t_{i^*|ii^*} + P^t_{i|ii^*}}\right| \geq \sqrt{\frac{2\lambda(P^t_{i^*|ii^*} - P^t_{i|ii^*})}{R_{ii^*}(t)\cdot(P^t_{i^*|ii^*} + P^t_{i|ii^*})}} + \frac{2\lambda(P^t_{i^*|ii^*} - P^t_{i|ii^*})}{3R_{ii^*}(t)\cdot(P^t_{i^*|ii^*} + P^t_{i|ii^*})}) \\ \leq 4\log(t)e^{\lambda} + 8e^{-\lambda/4}\,.\end{aligned}$$

Using the definition of $\Delta^t_{i^*i}$ and taking the union bound over all t and i gives us the desired bound.

The second part of the lemma (Equation 5.7.13) under the MNL model, follows from Equation 5.7.18, the fact that $R_{i|i^*}(t) = M_{i^*i}(t)\cdot(P^t_{i^*|i^*i} - P^t_{i|i^*i})$, and the fact that

$$\frac{P^t_{i^*|i^*i} - P^t_{i|i^*i}}{P^t_{i^*|i^*i} + P^t_{i|i^*i}} = \Delta^{\text{MNL}}_{i^*i}$$

We will now prove the third part of the lemma (Equation 5.7.14) under the MNL model for i, j such that $w_{i^*} - w_j \leq w_j - w_i$. Under this condition we have that

$$\begin{aligned} R_{ij}(t) &= \sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \subseteq S_{t'}] \cdot \frac{w_{i^*} - w_i}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a} \\ &\leq \sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \subseteq S_{t'}] \cdot 2 \cdot \frac{w_j - w_i}{\sum_{a \in S_{t'}} w_a} = 2M_{ij}(t) \cdot (P^t_{j|ij} - P^t_{i|ij}) \end{aligned}$$

Using Equation 5.7.18 and the above we get that

$$\begin{aligned} \Pr(\left| \frac{\widehat{P}^t_{i|ij}}{\widehat{P}^t_{j|ij} + \widehat{P}^t_{i|ij}} - \frac{w_i}{w_i + w_j} \right| \geq \sqrt{\frac{2\lambda(P^t_{j|ij} - P^t_{i|ij})}{R_{ij}(t) \cdot (P^t_{j|ij} + P^t_{i|ij})}} + \frac{2\lambda(P^t_{j|ij} - P^t_{i|ij})}{3R_{ij}(t) \cdot (P^t_{j|ij} + P^t_{i|ij})}) \\ \leq 4\log(t)e^{\lambda} + 8e^{-\lambda/4}. \end{aligned}$$

Using the definition of Δ^{MNL}_{ji} and taking the union bound over all t and i, j gives us the desired bound.

□

Recall that $N_{ij}(t)$ (defined in Equation 5.4.1) denotes the number of times (up to round t) that either arm i or j was chosen when they are played together and arm j is the anchor, and $M_{ij}(t)$ (defined in Equation 5.7.2) denotes the number of times i and j are played together when j is the anchor up to trial t. We will now prove a relation between N_{ij} and M_{ij} that is needed in the proof of our regret bounds.

Lemma 5.7.3 (Concentration of N_{ii^*})**.** *Let $S_1, \cdots, S_T$ be a sequence of subsets of $[n]$ and $y_1, \cdots, y_T$ be a sequence of choices according to this model, let $\mathcal{F}_t = \{S_1, y_1, \cdots, S_t, y_t\}$ be a filtration such that S_{t+1} is a measurable function of $\mathcal{F}_t$. For any $t \in [T]$ and any $i \in [n]$, we have that*

$$\Pr\left(N_{ii^*}(t) < \frac{(P^t_{i^*|ii^*} + P^t_{i|ii^*})}{2} \cdot M_{ii^*}(t), M_{ii^*}(t) \geq \frac{512\log(nCT)}{(P^t_{i*|ii^*} - P^t_{i|ii^*}) \cdot \Delta_{i^*i}} \right) \leq \frac{1}{(nT)^{30}}$$

where the quantities $\widehat{P}^t_{i|ii^*}$, $\widehat{P}^t_{i^*|ii^*}$, Δ_{i^*i} and $M_{ii^*}(t)$ are defined in Section 5.7.2.1. Moreover, if the underlying model is MNL, then for any $i, j \in [n]$ and any $t \in [T]$, we have that

$$\Pr\left(N_{ij}(t) \geq \frac{(P^t_{i|ij} + P^t_{j|ij})}{2} \cdot M_{ij}(t), M_{ij}(t) \geq \frac{512\log(nCT)}{(P^t_{j|ij} - P^t_{i|ij}) \cdot \Delta^{\text{MNL}}_{ji}}\right) \leq \frac{1}{(nT)^{30}}\,,$$

where the quantities $\widehat{P}^t_{i|ij}$, $\widehat{P}^t_{i|ij}$, Δ_{ji} and $M_{ij}(t)$ are defined in Section 5.7.2.1

Proof. Let us define X_t to be an indicator random variable denoting the event that either arm i or i^* won in trial t when i^* was the anchor, i.e. $X_t := \mathbb{1}[y_t \in \{i, i^*\}, a_t = i^*, \{i, i^*\} \subseteq S_t]$. Also, let Y_t be an indicator random variable denoting the event that i was played in trial t and i^* was the anchor arm, i.e. $Y_t := \mathbb{1}[a_t = i^*, \{i, i^*\} \subseteq S_t]$. Note that $M_{ii^*}(t) = \sum_{t'=1}^{t} Y_{t'}$ and $N_{ii^*}(t) = \sum_{t'=1}^{t} X_{t'}$. Throughout this proof we will let $P_{\{i,i^*\}|S_{t'}} := P_{i|S_{t'}} + P_{i^*|S_{t'}}$. Let $\alpha_t := (P^t_{i^*|ii^*} + P^t_{i|ii^*})$. and $\beta_t := \frac{512\log(nCT)}{(P^t_{i*|ii^*} - P^t_{i|ii^*})\cdot\Delta_{i^*i}}$ We also define Z_t as follows:

$$Z_t := \sum_{t'=1}^{t}\left(X_{t'} - P_{\{i,i^*\}|S_{t'}} \cdot Y_{t'}\right).$$

We can now write the deviation in N_{ii^*} in terms of the deviation in Z_t as follows:

$$\begin{aligned}\Pr\left(N_{ii^*}(t) < \frac{\alpha_t}{2} \cdot M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right) &= \Pr\left(\sum_{t'=1}^{t} X_{t'} < \sum_{t'=1}^{t} \frac{P_{\{i,i^*\}|S_{t'}}}{2} \cdot Y_{t'}, M_{ii^*}(t) \geq \beta_t\right)\\ &= \Pr\left(Z_t < -\sum_{t'=1}^{t} \frac{P_{\{i,i^*\}|S_{t'}}}{2} \cdot Y_{t'}, M_{ii^*}(t) \geq \beta_t\right)\\ &\leq \Pr\left(|Z_t| > \frac{\alpha_t}{2} \cdot M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right)\,,\end{aligned} \tag{5.7.19}$$

where the first equality follows from the definition of α_t given above and the definition of $P^t_{i^*|ii^*}$ and $P^t_{i|ii^*}$ given in Equation 5.7.3. Similar to the proof of Lemma 5.7.2, we will show that the random variables $\{Z_t\}$ form a martingale sequence with respect to the filtration

$\mathcal{F}_{t-1}$. To see this we will calculate $\mathbf{E}[Z_t|\mathcal{F}_{t-1}]$ as follows

$$\begin{aligned}\mathbf{E}[Z_t|\mathcal{F}_{t-1}] &= \mathbf{E}[Z_{t-1}|\mathcal{F}_{t-1}] + \mathbf{E}\Big[X_t - P_{\{i,i^*\}|S_{t'}} \cdot Y_t \Big| \mathcal{F}_{t-1}\Big] \\ &= Z_{t-1} + \mathbf{E}\Big[X_t \Big| \mathcal{F}_{t-1}\Big] - P_{\{i,i^*\}|S_{t'}} \cdot Y_t \,.\end{aligned}$$

The second equality holds because Y_t is a deterministic quantity given $\mathcal{F}_{t-1}$. In the case $Y_t = 0$ we have that $X_t = 0$; in the case that $Y_t = 1$, X_t is a Bernoulli random variable with probability $P_{\{i,i^*\}|S_{t'}}$. Hence, in both cases we have that $\mathbf{E}[X_t|\mathcal{F}_{t-1}] - P_{\{i,i^*\}|S_{t'}} \cdot Y_t = 0$. This implies that

$$\mathbf{E}[Z_t|\mathcal{F}_{t-1}, Y_t] = Z_{t-1} \,.$$

Hence, we have shown that the sequence Z_t's form a martingale sequence. We can now use the Bernstein inequality for martingales (Cesa-Bianchi and Lugosi, 2006) given in Theorem A.2.1 to bound the probability in Equation 5.7.15. This inequality bounds the deviation in Z_t using information about the second moments of the sequence. Let

$$\sigma_t^2 := \sum_{t'=1}^{t} \mathbf{E}\Big[(X_{t'} - P_{\{i,i^*\}|S_{t'}} \cdot Y_{t'})^2 | \mathcal{F}_{t'-1}\Big] \,.$$

We now calculate the value of σ_t^2. Recall that if $Y_t = 0$ then $X_t = 0$; and if $Y_t = 1$ then X_t is a Bernoulli random variable with probability $P_{\{i,i^*\}|S_{t'}}$. We then have that

$$\begin{aligned}\sigma_t^2 &= \sum_{t'=1}^{t} \mathbf{E}\Big[(X_{t'} - P_{\{i,i^*\}|S_{t'}} \cdot Y_{t'})^2 | \mathcal{F}_{t'-1}\Big] \\ &= \sum_{t'=1}^{t} Y_{t'} \cdot \mathrm{Var}(X_{t'}|\mathcal{F}_{t'-1}, Y_{t'} = 1) + (1 - Y_{t'}) \cdot 0 \\ &= \sum_{t'=1}^{t} Y_{t'} \cdot P_{\{i,i^*\}|S_{t'}}(1 - P_{\{i,i^*\}|S_{t'}}) \leq \alpha_t M_{ii^*}(t) \,. \end{aligned} \tag{5.7.20}$$

We then have

$$\Pr(|Z_t| > \alpha_t M_{ii^*}(t)/2, M_{ii^*}(t) \geq \beta_t) \leq \Pr(|Z_t| \geq \alpha_t M_{ii^*}(t)/2, \sigma_t^2 \leq \alpha_t M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t) \,.$$

Also, $|X_t - P_{\{i,i^*\}|S_{t'}} \cdot Y_t| \leq 1$. Using the Bernstein's inequality for martingales, we have that,

$$\Pr\left(|Z_t| > \sqrt{2\nu\lambda} + 2\lambda/3, \sigma_t^2 \leq \nu\right) \leq 2e^{-\lambda}\,,$$

for any constants $\lambda, \nu > 0$. However, the problem with our desired bound is that we want to bound the deviation of Z_t by a quantity that depends on $\alpha_t M_{ii^*}(t)$ which is a random variable, whereas in the above inequality we need λ and ν to be constants. We use the peeling technique (Bartlett et al. (2005)), and break down the process into different variance classes. We will then take a union bound over all the variance classes, i.e. values of $\alpha_t M_{ii^*}(t)$.

Let us define $f(\nu, \lambda) = \sqrt{2\nu\lambda} + 2\lambda/3$ for any ν, λ. Let $\lambda = \alpha_t M_{ii^*}(t)/16$. We are interested in the events where $M_{ii^*}(t) \geq \beta_t$, i.e. $\alpha_t M_{ii^*}(t) \geq \alpha_t\beta_t \geq 512\log(nCT)$. This implies that $\lambda \geq 32\log(nCT)$. We then have

$$\begin{aligned}
&\Pr\left(|Z_t| \geq \alpha_t M_{ii^*}(t)/2, \sigma_t^2 \leq \alpha_t M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right)\\
&\quad\leq \Pr\left(|Z_t| > f(\alpha_t M_{ii^*}(t), \lambda)\,,\, \sigma_t^2 \leq \alpha_t M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right)\\
&\quad\leq \sum_{r=1}^{\lceil\log(t)\rceil} \Pr\left(\frac{t}{2^r} < \alpha_t M_{ii^*}(t) \leq \frac{t}{2^{r-1}}, |Z_t| > f(\alpha_t M_{ii^*}(t), \lambda), \sigma_t^2 \leq \alpha_t M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right)\\
&\quad\leq \sum_{r=1}^{\lceil\log(t)\rceil} \Pr\left(|Z_t| > f(\frac{t}{2^r}, 32\log(nCT))\,,\, \sigma_t^2 \leq \frac{t}{2^{r-1}}\right)
\end{aligned}$$

.

The second last inequality is due to the union bound. We now use the Bernstein's inequality to bound the above as:

$$\begin{aligned}
&\Pr\left(|Z_t| > f(\gamma, 32\log(nCT))\,, \sigma_t^2 \leq 2\gamma\right)\\
&\quad\leq \Pr\left(|Z_t| > \sqrt{128\gamma\log(nCT)} + 64\log(nCT)/3\,, \sigma_t^2 \leq 2\gamma\right)\\
&\quad\leq 2e^{-32\log(nCT)}\,.
\end{aligned}$$

Combining this all together, we have that

$$\Pr\left(|Z_t| > \frac{\alpha_t}{2} \cdot M_{ii^*}(t), M_{ii^*}(t) \geq \beta_t\right) \leq \frac{2\log T}{(nCT)^{32}}\,.$$

Using the above and applying the union bound over all arms and trails gives the desired bound. The proof of the MNL case follows the same argument with i^* replaced with j. $\square$

5.7.2.3 Proof of Regret Upper Bound for WBA-A (Theorem 5.5.1)

In this section we will prove the regret bound for our WBA-A algorithm given in Theorem 5.5.1. The proof of this theorem hinges on three main lemmas given below. A flow-chart for the proof is given in Figure 13. Before stating these lemmas, we would like to remind the reader that the execution of our algorithm is divided in rounds and each round contain up to n trials. The first lemma bounds the number of rounds arm i^* is not in the active set.

Lemma 5.7.4 (Number of rounds where i^* is not active)**.** *Fix an anchor arm $a \in [n] \setminus \{i^*\}$. The expected number of rounds arm i^* will not be a part of the active set is bounded as*

$$\mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[i^* \notin A_r]\right] \leq 2\,.$$

The proof of this lemma is given in Section 5.7.2.5. We will define a_r to be the arm that empirically beats all other arms at the end of round $r-1$ if such an arm exists, i.e. $\sum_{j\in[n]} \mathbb{1}[\widehat{P}_{ja_r}(t) \leq \frac{1}{2}] = n-1$, where t is the last trial in round $r-1$. If there is no arm that empirically beats all other arms then we will let $a_r = 0$. If there are multiple such arms, then we will choose one arbitrarily. The following lemma will now bound the number of rounds arm i^* does not empirically beat every other arm.

Lemma 5.7.5 (Time when i^* is not the empirically best arm)**.** *The total number of rounds when the best arm i^* will not be the empirically best arm, even when it is in the active set, is* *upper bounded as*

$$\mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[a_r \neq i^*, i^* \in A_r]\right] \leq \sum_{i \in [n] \setminus \{i^*\}} \frac{1}{\exp d(1/2, P^{\mathrm{GCC}}_{i^*i}) - 1},$$

where $P^{\mathrm{GCC}}_{i^*i}$ is defined in Equation 5.5.2.

The proof of this lemma is given in Section 5.7.2.6. Note that if $a_r = i^*$ then the anchor arm in all the trials in that round becomes i^*. The following lemma now bounds the regret incurred due to each suboptimal arm when played against the anchor i^*.

Lemma 5.7.6 (Regret due to a bad arm)**.** *Given an arm $i \in [n] \setminus \{i^*\}$ the expected regret incurred due to arm i when arm i^* is the anchor is upper bounded as*

$$\mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = i^*, i \in S_t]\right] \leq \frac{512 \log(nCT)}{\Delta_{i^*i}} + 13,$$

where Δ_{i^*i} is defined in Equation 5.3.1 and $r(S_t, i)$ is defined in Equation 5.7.5.

The proof of this lemma is given in Section 5.7.2.7. We will now prove Theorem 5.5.1 using the three lemmas above.

Proof of Theorem 5.5.1. The execution of the algorithm can roughly be divided into three intermittent phases– (1) when the GCW arm i^* is not in the active set, (2) when i^* is in the active set but does not beat all other arms empirically, i.e. $a_r \neq i^*$, (3) when i^* is in the active set and also beats all other arms empirically. The three lemmas above bound the number of rounds spent in these three phases.

However, in order to prove a regret upper bound we will also have to bound the total regret incurred due to a single round. The first thing to observe is that each arm is played at most once in each round except a few arms that might be played multiple times due to step 6 of the algorithm. Hence, the regret for all steps except step 6 is upper bounded by n as the regret for each arm is at most 1. Now, in order to bound the regret for step 6, we need to observe that the number of times the anchor arm is changed in a single round can be at most $\log n$. This is due to the fact that $A_r \setminus Q$ reduces by a factor of at least 2 each time a new anchor arm is selected by the algorithm. Now, we can bound the regret incurred due to step 6 of the algorithm by $k \log n \leq n \log n$ as the regret for each arm is upper bounded by 1 and there can be at most k arms added in step 6 per anchor arm.

Hence, we now have that

$$\begin{aligned}
\mathbf{E}[R(T)] &\leq n \log n \cdot \left(\mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[i^* \notin A_r] \right] + \mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[a_r \neq i^*, i^* \in A_r] \right] \right) \\
&\qquad + \sum_{i \in [n] \setminus \{i^*\}} \mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = i^*, i \in S_t] \right] \\
&\leq n \log n \cdot \left(2 + \sum_{i \in [n] \setminus \{i^*\}} \frac{1}{\exp d(1/2, P_{i^*i}^{\text{GCC}}) - 1} \right) + \sum_{i \in [n] \setminus \{i^*\}} \left(\frac{512 \log(nCT)}{\Delta_{i^*i}} + 13 \right) \\
&\leq n \log n \cdot \left(2 + \frac{n}{\Delta_{\min}^2} \right) + 13n + \sum_{i \in [n] \setminus \{i^*\}} \frac{512 \log(n)}{\Delta_{i^*i}} + \sum_{i \in [n] \setminus \{i^*\}} \frac{512 \log(CT)}{\Delta_{i^*i}} \\
&= O\left(\frac{n^2 \log n}{\Delta_{\min}^2} \right) + O\left(\sum_{i \in [n] \setminus i^*} \frac{\log(TC)}{\Delta_{i^*i}} \right)
\end{aligned}$$

where the last inequality follows from the fact that $\exp\{d(1/2, P_{i^*i}^{\text{GCC}})\} - 1 \geq d(1/2, P_{i^*i}^{\text{GCC}}) \geq 2(P_{i^*i}^{\text{GCC}} - \frac{1}{2})^2 = \Delta_{\min}^2/2$ which follows using the well-known Pinsker's inequality. This gives the desired bound under any GCC model.

Now, if the underlying GCC model is MNL, using the defintion of $\Delta_{i^*i}^{\text{MNL}}$ and $\Delta_{\min}^{\text{MNL}}$ we easily have

$$\mathbf{E}[R(T)] \leq O\left(\frac{n^2 \log n}{(\Delta_{\min}^{\text{MNL}})^2} \right) + O\left(\sum_{i \in [n] \setminus i^*} \frac{\log(TC)}{\Delta_{i^*i}^{\text{MNL}}} \right).$$

□

5.7.2.4 Proof of Regret Upper Bound for WBA-L (Theorem 5.5.2)

In this section we will prove the regret upper bound for our WBA-L algorithm given in Theorem 5.5.2. The proof of this theorem hinges on three main lemmas. Figure 13 gives a flow-chart depicting the various lemmas involved in this proof. The first lemma will bound the number of rounds where i^* is not active.

Lemma 5.7.4 (Number of rounds where i^* is not active). Fix an anchor arm $a \in [n] \setminus \{i^*\}$. The expected number of rounds arm i^* will not be a part of the active set is bounded as

$$\mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[i^* \notin A_r]\right] \leq 2\,.$$

This is the same lemma that is used in the proof of Theorem 5.5.1 and the proof of this lemma is given in Section 5.7.2.5. We will also like to remind the reader that an anchor arm a_r is selected in each round which is a considered the candidate best arm by the algorithm. Recall that under the MNL model there is a total ordering σ over the arms such that $\sigma_i < \sigma_j$ if $w_i < w_j$. Let us also define the event $\mathcal{E}_\lambda$ as follows:

$$\begin{aligned}\mathcal{E}_\lambda := \Bigg\{ & \left|\frac{\widehat{P}^t_{i|ij}}{\widehat{P}^t_{j|ij} + \widehat{P}^t_{i|ij}} - \frac{w_i}{w_i + w_j}\right| \leq \sqrt{\frac{4\Delta^{\text{MNL}}_{ji}\lambda}{R_{ij}(t)}} + \frac{4\Delta^{\text{MNL}}_{ji}\lambda}{3R_{ij}(t)} \\ & \quad \forall i, j \in [n] \text{ s.t. } w_{i^*} - w_j \leq w_j - w_i\,, \\ & \text{and } \left|\frac{\widehat{P}^t_{i|i^*i}}{\widehat{P}^t_{i|i^*i} + \widehat{P}^t_{i^*|i^*i}} - \frac{w_i}{w_i + w_{i^*}}\right| \leq \sqrt{\frac{2\Delta^{\text{MNL}}_{i^*i}\lambda}{R_{ii^*}(t)}} + \frac{2\Delta^{\text{MNL}}_{i^*i}\lambda}{3R_{ii^*}(t)}, \quad \forall i \in [n]\,, \\ & \text{and } \left(N_{ij}(t) \geq \frac{(P^t_{i|ij} + P^t_{j|ij})}{2} \cdot M_{ij}(t), M_{ij}(t) \geq \frac{512\log(nCT)}{(P^t_{j|ij} - P^t_{i|ij}) \cdot \Delta^{\text{MNL}}_{ji}}\right)\Bigg\}\end{aligned}$$

We will define a mistake-free execution of WBA-L.

Definition 5.7.7 (Mistake-free execution). *We say a* mistake *is made in the execution of* WBA-L *if for some r, $w_{a_r} < w_{a_{r+1}}$. We will say a call to* WBA-L *is* mistake-free *if it makes* *no mistake.*

The next lemma bounds the probability of an arm becoming an anchor arm.

Lemma 5.7.8 (Probability of becoming anchor)**.** *Given a set of arms $[n]$, and MNL weights $\{w_i\}_{i\in[n]}$, let $(\sigma_1, \cdots, \sigma_n)$ be an ordering of arms such that $\sigma_i \in [n]$ is the arm at position i and $w_{\sigma_i} \geq w_{\sigma_{i+1}}$ for $i \in [n]$. Let X_i be a random variable indicating that arm σ_i becomes an anchor arm for some round. If the execution of* WBA *is mistake-free, then we have*

$$\Pr(X_j = 1) \leq \frac{1}{j}\,.$$

The proof of this lemma is similar to the proof of a similar bound shown in Yue et al. (2009) for the IF algorithm and is given in Section 5.7.2.8 below. We will denote by $\mathcal{T}(r)$ all the trails that belong to round r, i.e. if round r starts at trial t and ends at $t' \geq t$ then $\mathcal{T}(r) := \{t, t+1, \cdots, t'\}$. For $t \in \mathcal{T}(r)$ we will also denote by a_t the anchor arm that was selected at the beginning of round r, and by A_t we will denote the set of active arms at the beginning of round r. The last lemma bounds the regret for any given anchor arm.

Lemma 5.7.9 (Regret due to a bad arm)**.** *Given an arm $i \in [n] \setminus \{i^*\}$, the expected regret incurred due to arm i when arm j is the anchor conditional on event $\mathcal{E}_\lambda$ for $\lambda := 8\log(nCT)$, is upper bounded as*

$$\mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = j, i \in S_t] \;\middle|\; \mathcal{E}_\lambda\right] \leq \Pr(\exists t \in [T] : a_t = j | \mathcal{E}_\lambda) \cdot \left(\frac{2048\log(nCT)}{\Delta^{\mathrm{MNL}}_{i^*i}} + 1\right) + \mathbf{E}\left[\sum_{t=1}^{T} \mathbb{1}[a_t = j, i \in S_t, i^* \notin A_t] \;\middle|\; \mathcal{E}_\lambda\right],$$

where $\Delta^{\mathrm{MNL}}_{i^*i}$ is defined in Equation 5.3.1 and $r(S_t, i)$ is defined in Equation 5.7.5.

The proof of this lemma is given in Section 5.7.2.9. We are now ready to prove Theorem 5.5.2.

Proof. Fix $\lambda := 8\log(nCT)$. We have that

$$\mathbf{E}[R(T)] = \Pr(\mathcal{E}_\lambda) \cdot \mathbf{E}\left[R(T)|\mathcal{E}_\lambda\right] + (1 - \Pr(\mathcal{E}_\lambda)) \cdot \mathbf{E}\left[R(T)|\neg\mathcal{E}_\lambda\right] . \tag{5.7.21}$$

We will now bound each of the terms in the above equation one by one. We first have

$$\mathbf{E}\left[R(T)|\neg\mathcal{E}_\lambda\right] \leq kT \,, \tag{5.7.22}$$

which follows from the fact that the $R_{ii^*}(t)$ is upper bounded by 1 in each trial t and there are at most T trials. Also, using Lemma 5.7.2 and Lemma 5.7.3 we can observe that the event $\mathcal{E}_\lambda$ happens with high probability, i.e.

$$1 - \Pr(\mathcal{E}_\lambda) \leq \frac{25}{nT} \,. \tag{5.7.23}$$

Combining the above gives a bound on the second quantity of Equation 5.7.21. Now, we will bound the first quantity in Equation 5.7.21. Without loss of generality, assume that the arms are ordered such that $w_1 > w_2 \geq w_3 \cdots \geq w_n$. We have that

$$\begin{aligned}
\mathbf{E}\left[R(T)|\mathcal{E}_\lambda\right] &= \sum_{j\in[n]} \sum_{i\in[n]} \mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = j, i \in S_t] \;\middle|\; \mathcal{E}_\lambda\right] \\
&\leq \sum_{j\in[n]} \sum_{i\in[n]} \Pr(\exists t \in [T] : a_t = j|\mathcal{E}_\lambda) \cdot \left(\frac{2048\log(nCT)}{\Delta^{\text{MNL}}_{i^*i}} + 1\right) \\
&\quad + \sum_{j\in[n]} n \cdot \mathbf{E}\left[\sum_{t=1}^{T} \mathbb{1}[a_t = j, i \in S_t, i^* \notin A_t] \;\middle|\; \mathcal{E}_\lambda\right] \\
&\leq \sum_{j\in[n]} \Pr(\exists t \in [T] : a_t = j|\mathcal{E}_\lambda) \cdot \left(\sum_{i\in[n]} \frac{2048\log(nCT)}{\Delta^{\text{MNL}}_{i^*i}} + n\right) \\
&\quad + n \cdot \mathbf{E}\left[\sum_{t=1}^{T} \mathbb{1}[i \in S_t, i^* \notin A_t] \;\middle|\; \mathcal{E}_\lambda\right] ,
\end{aligned}$$

where the first inequality follows due to Lemma 5.7.9. We first have that

$$\mathbf{E}\left[\sum_{t=1}^{T} \mathbb{1}[i \in S_t, i^* \notin A_t] \,\middle|\, \mathcal{E}_\lambda\right] \leq 2\,,$$

using Lemma 5.7.4. Now, observe that given the event $\mathcal{E}_\lambda$ occurs we will have concentration for all trials and arms, and hence, an arm which is worse than the current anchor will not be able to replace the anchor. Hence, similar to Yue et al. (2009), this implies that the execution of WBA-L will be mistake-free. Hence, using Lemma 5.7.8 we have that

$$\Pr(\exists t \in [T] : a_t = j | \mathcal{E}_\lambda) \leq \frac{1}{j}\,.$$

Combining the above inequalities we have that

$$\mathbf{E}\left[R(T)|\mathcal{E}_\lambda\right] \leq \sum_{j \in [n]} \frac{1}{j} \cdot \left(\sum_{i \in [n] \setminus \{i^*\}} \frac{2048 \log(nCT)}{\Delta_{i^*i}^{\mathrm{MNL}}} + n \right) + 2n \tag{5.7.24}$$

$$\leq \log(n) \left(\sum_{i \in [n] \setminus \{i^*\}} \frac{2048 \log(nCT)}{\Delta_{i^*i}^{\mathrm{MNL}}} + n \right) + 2n \tag{5.7.25}$$

Combining Equation 5.7.21 and Equation 5.7.25 gives the required bound. □

5.7.2.5 Proof of Lemma 5.7.4

Proof. We have that

$$\mathbf{E}\left[\sum_{r=1}^{T} \mathbb{1}[i^* \notin A_r]\right] = \mathbf{E}\left[\sum_{r=2}^{T} \mathbb{1}[i^* \notin A_r]\right] \leq \mathbf{E}\left[\sum_{t=2}^{T} \mathbb{1}[\neg \mathcal{J}_{i^*}(t, C)]\right]\,.$$

The first equality above follows due to the fact that A_1 will always include i^*. Using the union bound we have the following inequality-

$$\begin{aligned}\mathbb{1}[\neg\mathcal{J}_{i^*}(t,C)] \leq \sum_{S\subseteq[n]\setminus\{i^*\}} & \sum_{\{n_a\}\in[T]^S}\sum\cdots\sum \\ & \mathbb{1}[\bigcap_{a\in S}\{N_{i^*a}(t)=n_a, \widehat{P}_{i^*a}(t)<\frac{1}{2}\}\cap\bigcap_{a\notin S}\{\widehat{P}_{i^*a}(t)\geq\frac{1}{2}\}\cap\{\neg\mathcal{J}_{i^*}(t,C)\}]\,.\end{aligned}$$

Fix some set $S\subseteq[n]\setminus\{i^*\}$. Also, let $s:=|S|$. Fix some $n_a\in[T]$ for all $a\in S$. Let $\widehat{P}^{n_a}_{i^*a}$ be the empirical probability of i^* beating a after being pulled together n_a times. We will analyze the number of rounds that i^* is excluded from the active set due to the above configuration of $S,\{n_a\}$. The conditions $\mathcal{J}_{i^*}(t,C)$ will hold when

$$\sum_{a\in S} n_a d(\widehat{P}^{n_a}_{i^*a},\frac{1}{2}) \leq \log(t)+s\log(nC) \implies t\geq \exp\left(\sum_{a\in S} n_a d(\widehat{P}^{n_a}_{i^*a},\frac{1}{2}) - s\log(nC)\right)\,.$$

Hence, we have that

$$\begin{aligned}&\sum_{t=2}^{\infty}\mathbb{1}[\bigcap_{a\in S}\{N_{i^*a}(t)=n_a,\widehat{P}_{i^*a}(t)<\frac{1}{2}\}\cap\bigcap_{a\notin S}\{\widehat{P}_{i^*a}(t)\geq\frac{1}{2}\}\cap\{\neg\mathcal{J}_{i^*}(t,C)\}] \\ &\qquad \leq \exp\left(\sum_{a\in S} n_a d(\widehat{P}^{n_a}_{i^*a},\frac{1}{2}) - s\log(nC)\right)\,.\end{aligned}$$

Now, we will use the method similar to the one used in Lemma 5 of Komiyama et al. (2015b), to bound the expectation of the above quantity. Fix $x_a\in[0,\log 2]$ for all $a\in S$. Let $P_a(x_a)=\Pr\left(\ \widehat{P}^{n_a}_{i^*a}\leq\frac{1}{2}, d^+(\widehat{P}^{n_a}_{i^*a},\frac{1}{2})\geq x_a\ \right)$, where $d^+(P,Q)=\mathbb{1}[P\leq Q]\cdot d(P,Q)$. We then have

$$
\begin{aligned}
&\mathbf{E}\left[\sum_{t=2}^{T} \mathbb{1}[\bigcap_{a\in S}\{N_{i^*a}(t) = n_a, \widehat{P}_{i^*a}(t) < \frac{1}{2}\} \cap \bigcap_{a\notin S}\{\widehat{P}_{i^*a}(t) \geq \frac{1}{2}\} \cap \{\neg \mathcal{J}_{i^*}(t, C)\}]\right] \\
&\quad \leq \int_{\{x_a\}\in[0,\log(2)]^{|S|}} \exp\left(\sum_{a\in S} n_a x_a - s\log(nC)\right) \prod_{a\in S} \mathrm{d}(-P_a(x_a)) \\
&\quad = \exp\left(-s\log(nC)\right) \cdot \prod_{a\in S} \int_{x_a\in[0,\log(2)]} \exp\left(n_a x_a\right) \mathrm{d}(-P_a(x_a))
\end{aligned}
$$

(due to the independence of comparisons with respect to different anchors)

$$
\begin{aligned}
&\quad = \exp\left(-s\log(nC)\right) \cdot \\
&\qquad \prod_{a\in S}\left([-\exp(n_a x_a) P_a(x_a)]_0^{\log(2)} + \int_{x_a\in[0,\log(2)]} n_a \exp\left(n_a x_a\right) P_a(x_a) \mathrm{d}x_a \right)
\end{aligned}
$$

(integration by parts)

$$
\begin{aligned}
&\quad \leq \exp\left(-s\log(nC)\right) \cdot \\
&\qquad \prod_{a\in S}\left(P_a(0) + \int_{x_a\in[0,\log(2)]} n_a \exp\left(n_a x_a\right) \exp -n_a(x_a + C_1(P^{\mathrm{GCC}}_{i^*a}, \frac{1}{2}))\mathrm{d}x_a \right)
\end{aligned}
$$

(Using Lemma 5.5.1, Fact 10 in Komiyama et al. (2015b) with $C_1(p,q) = (p-q)^2/2p(1-q)$)

$$
\begin{aligned}
&\quad = \exp\left(-s\log(nC)\right) \cdot \\
&\qquad \prod_{a\in S}\left(\exp -n_a d(\frac{1}{2}, P^{\mathrm{GCC}}_{i^*a}) + \int_{x_a\in[0,\log(2)]} n_a \exp -n_a C_1(P^{\mathrm{GCC}}_{i^*a}, \frac{1}{2})\mathrm{d}x_a \right) \\
&\quad = \exp\left(-s\log(nC)\right) \cdot \prod_{a\in S}\left(\exp -n_a d(\frac{1}{2}, P^{\mathrm{GCC}}_{i^*a}) + \log(2) n_a \exp -n_a C_1(P^{\mathrm{GCC}}_{i^*a}, \frac{1}{2}) \right) .
\end{aligned}
$$

We will now take a union bound over $\{n_a\}$. We have that

$$\begin{aligned}\sum\sum_{\{n_a\}\in[T]^S}\cdots\sum \exp\left(-s\log(nC)\right)\cdot\prod_{a\in S}\left(\exp -n_a d(\frac{1}{2},P^{\text{GCC}}_{i^*a}) + \log(2)n_a\exp -n_a C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\right)\\ = \exp\left(-s\log(nC)\right)\cdot\\ \prod_{a\in S}\sum_{n_a}\left(\exp -n_a d(\frac{1}{2},P^{\text{GCC}}_{i^*a}) + \log(2)n_a\exp -n_a C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\right)\\ \leq \exp\{-s\log(nC)\}\cdot\prod_{a\in S}\left(\frac{1}{\exp d(\frac{1}{2},P^{\text{GCC}}_{i^*a})-1}+\frac{\exp\{C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\}}{(\exp\{C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\}-1)^2}\right)\\ \leq \exp\{-s\log(nC)+s\log(C')\}\,,\end{aligned}$$

where the constant C' is defined as

$$C' := \max_{a\in[n]\setminus i^*}\left(\frac{1}{\exp d(\frac{1}{2},P^{\text{GCC}}_{i^*a})-1}+\frac{\exp\{C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\}}{(\exp\{C_1(P^{\text{GCC}}_{i^*a},\frac{1}{2})\}-1)^2}\right)\,.$$

We will now apply the union bound over all subsets $S\subseteq[n]\setminus i^*$. Now, if the parameter C is larger than C', then we have

$$\begin{aligned}\sum_{S\subseteq[n]\setminus\{i^*\}}\exp\{-|S|\log(nC)+|S|\log(C')\} = \sum_{s=1}^{n-1}\sum_{S\subseteq[n]\setminus\{i^*\},|S|=s}\exp -s\log(nC)+s\log(C')\\ \leq \sum_{s=1}^{n-1}\left(\frac{en}{s}\right)^s\exp -s\log(nC)+s\log(C')\\ = \sum_{s=1}^{n-1}\exp -s\log(nC)+s\log(C')+s\log(n)+s-s\log(s)\\ \leq \sum_{s=1}^{n-1}\exp s - s\log(s)\leq 2\,.\end{aligned}$$

□

5.7.2.6 Proof of Lemma 5.7.5

Proof. In the following we overload notation slightly and for a round r define $N_{ii^*}(r)$ and $\widehat{P}_{ii^*}(r)$ to be the equal to $N_{ii^*}(t)$ and $\widehat{P}_{ii^*}(t)$, where t is the last trial in round r. We have the following set of inequalities:

$$\begin{aligned}
\mathbf{E}\left[\sum_{r=1}^{T}\mathbb{1}[a_r \neq i^*, i^* \in A_r]\right] &= \mathbf{E}\left[\sum_{r=1}^{T}\mathbb{1}[\exists i \neq i^*, i^* \in A_r, N_{ii^*}(r) > N_{ii^*}(r-1), \widehat{P}_{i^*i}(r-1) \leq \frac{1}{2}]\right] \\
&\leq \mathbf{E}\left[\sum_{r=1}^{T}\sum_{i\in[n]\setminus i^*}\mathbb{1}[i^* \in A_r, N_{ii^*}(r) > N_{ii^*}(r-1), \widehat{P}_{i^*i}(r-1) \leq \frac{1}{2}]\right] \\
&\leq \mathbf{E}\left[\sum_{r=1}^{T}\sum_{i\in[n]\setminus\{i^*\}}\sum_{n_i=0}^{T}\mathbb{1}[N_{ii^*}(r-1) = n_i, N_{ii^*}(r) > n_i, \widehat{P}_{i^*i}^{n_i} \leq \frac{1}{2}]\right] \\
&= \mathbf{E}\left[\sum_{i\in[n]\setminus\{i^*\}}\sum_{r=1}^{T}\sum_{n_i=0}^{T}\mathbb{1}[N_{ii^*}(r-1) = n_i, N_{ii^*}(r) > n_i, \widehat{P}_{i^*i}^{n_i} \leq \frac{1}{2}]\right] \\
&\leq \mathbf{E}\left[\sum_{i\in[n]\setminus\{i^*\}}\sum_{n_i=0}^{T}\mathbb{1}[\widehat{P}_{i^*i}^{n_i} \leq \frac{1}{2}]\right] \\
&= \sum_{i\in[n]\setminus\{i^*\}}\sum_{n_i=0}^{T}\mathbf{E}\left[\mathbb{1}[\widehat{P}_{i^*i}^{n_i} \leq \frac{1}{2}]\right] \\
&= \sum_{i\in[n]\setminus\{i^*\}}\sum_{n_i=0}^{T}\exp -n_i d(1/2, P_{i^*i}^{\text{GCC}}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(using concentration Lemma 5.5.1)} \\
&= \sum_{i\in[n]\setminus\{i^*\}}\frac{1}{\exp d(1/2, P_{i^*i}^{\text{GCC}}) - 1}
\end{aligned}$$

□

5.7.2.7 Proof of Lemma 5.7.6

Proof. In order to prove this lemma we will use the concentration lemmas given in Section 6.6, specifically Lemma 5.7.2 and Lemma 5.7.3. Let us define $\lambda := 8\log(nCT)$. Let us also define the event $\mathcal{E}_\lambda$ as follows:

$$\mathcal{E}_\lambda := \left\{ \left| \frac{\widehat{P}^t_{i|ii^*}}{\widehat{P}^t_{i^*|ii^*} + \widehat{P}^t_{i|ii^*}} - \frac{P^t_{i|ii^*}}{P^t_{i^*|ii^*} + P^t_{i|ii^*}} \right| \leq \sqrt{\frac{2\Delta^t_{i^*i}\lambda}{R_{ii^*}(t)}} + \frac{2\Delta^t_{i^*i}\lambda}{3R_{ii^*}(t)} \text{ and} \right. \tag{5.7.26}$$

$$\left. \left(N_{ii^*}(t) \geq \frac{(P^t_{i^*|ii^*} + P^t_{i|ii^*})}{2} \cdot M_{ii^*}(t), M_{ii^*}(t) \geq \frac{512\log(nCT)}{(P^t_{i*|ii^*} - P^t_{i|ii^*}) \cdot \Delta_{i^*i}} \right) \right\} \tag{5.7.27}$$

We then have that

$$\begin{aligned} \mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = i^*, i \in S_t]\right] &= \mathbf{E}\left[\sum_{t=1}^{T} R_{ii^*}(t)\right] \\ &= \Pr(\mathcal{E}_\lambda) \cdot \mathbf{E}\left[\sum_{t=1}^{T} R_{ii^*}(t)|\mathcal{E}_\lambda\right] \\ &\quad + (1 - \Pr(\mathcal{E}_\lambda)) \cdot \mathbf{E}\left[\sum_{t=1}^{T} R_{ii^*}(t)|\neg\mathcal{E}_\lambda\right]. \end{aligned} \tag{5.7.28}$$

We will now bound each of the terms in the above equation one by one. We first have

$$\mathbf{E}\left[\sum_{t=1}^{T} R_{ii^*}(t)|\neg\mathcal{E}_\lambda\right] \leq T\,, \tag{5.7.29}$$

which follows from the fact that the $R_{ii^*}(t)$ is upper bounded by 1 in each trial t and there are at most T trials. Also, using Lemma 5.7.2 and Lemma 5.7.3 we can observe that the event $\mathcal{E}_\lambda$ happens with high probability, i.e.

$$1 - \Pr(\mathcal{E}_\lambda) \leq \frac{12}{T}\,. \tag{5.7.30}$$

Combining the above gives a bound on the second quantity of Equation 5.7.28.

Let us define $R_{\max} := \frac{512\log(nCT)}{\Delta_{i^*i}} + 1$. Finally, we will argue that the expected regret $R_{ii^*}(t)$ conditional on the event $\mathcal{E}_\lambda$ is upper bounded as

$$\mathbf{E}\left[\sum_{t=1}^{T} R_{ii^*}(t)|\mathcal{E}_\lambda\right] \leq R_{\max} \tag{5.7.31}$$

Towards a contradiction, suppose that the above regret is strictly larger than $R_{\max}$ for some round s. Let $t' < s$ be the first round at which the regret becomes larger than or equal to $R_{\max} - 1$. Note that $R_{ii^*}(t') < R_{\max}$ since the regret before round t' is strictly less than $R_{\max} - 1$ by definition and the regret can at most increase by 1 each round. We will now show that for any round t after t', arm i will not be a part of the active set of arms, thereby leading to a contradiction. To see this observe that,

$$\sqrt{\frac{2\Delta^t_{i^*i}\lambda}{R_{\max}-1}} + \frac{2\Delta^t_{i^*i}\lambda}{3(R_{\max}-1)} \leq \sqrt{\frac{\Delta^t_{i^*i}\cdot\Delta_{i^*i}}{32}} + \frac{\Delta^t_{i^*i}\cdot\Delta_{i^*i}}{96} \leq \frac{\Delta^t_{i^*i}}{4},$$

where the above inequality follows from the fact that $\Delta_{i^*i} \leq \Delta^t_{i^*i} \leq 1$. Given that the event $\mathcal{E}_\lambda$, we have that

$$\left|\frac{\widehat{P}^t_{i|ii^*}}{\widehat{P}^t_{i^*|ii^*} + \widehat{P}^t_{i|ii^*}} - \frac{P^t_{i|ii^*}}{P^t_{i^*|ii^*} + P^t_{i|ii^*}}\right| < \frac{\Delta^t_{i^*i}}{4}.$$

Recall from Equation 5.4.2 that $\widehat{P}_{ii^*}(t) = \frac{\widehat{P}^t_{i|ii^*}}{\widehat{P}^t_{i^*|ii^*}+\widehat{P}^t_{i|ii^*}}$ and $P_{ii^*}(t) = \frac{P^t_{i^*|ii^*}}{P^t_{i^*|ii^*}+P^t_{i|ii^*}}$. Using the definition of $\Delta^t_{i^*i}$ we know that $\Delta^t_{i^*i} = 2(1/2 - P_{ii^*}(t))$. Using this we have that

$$|\widehat{P}_{ii^*}(t) - P_{ii^*}(t)| < \frac{\Delta^t_{i^*i}}{4} \implies \widehat{P}_{ii^*}(t) < P_{ii^*}(t) + \frac{\Delta^t_{i^*i}}{4} \implies \widehat{P}_{ii^*}(t) < \frac{1}{2} - \frac{\Delta^t_{i^*i}}{4}. \quad (5.7.32)$$

Using this bound, we will now argue that the above condition is sufficient to ensure that i will not be included in the active set A_t for any trials $t > t'$. To see this recall that in order to include i in the active set at time t we need $\mathcal{J}_i(t, C) = 0$ which is defined as:

$$\mathcal{J}_i(t, C) = \mathbb{1}\Big\{\exists S \subseteq [n] : I_i(t, S) \geq |S|\log(nC) + \log(t)\Big\},$$

where

$$I_i(t, S) = \sum_{j\in S} \mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2}).$$

Consider the set $S = \{i^*\}$. In order to show that $\mathcal{J}_i(t, C) = 1$ for all $t > t'$ we want to show that

$$\mathbb{1}[\widehat{P}_{ii^*}(t) \leq \frac{1}{2}] \cdot N_{ii^*}(t) \cdot d(\widehat{P}_{ii^*}(t), \frac{1}{2}) \geq \log(nCT)\,. \tag{5.7.33}$$

Using the well-known Pinsker's inequality we have that $d(P, Q) \geq 2(P-Q)^2$ for any $0 \leq P, Q \leq 1$. Combining this with Equation 5.7.32, we have that

$$\mathbb{1}[\widehat{P}_{ii^*}(t) \leq \frac{1}{2}] \cdot d(\widehat{P}_{ii^*}(t), \frac{1}{2}) \geq 2(\widehat{P}_{ii^*}(t) - \frac{1}{2})^2 > \frac{\left(\Delta^t_{i^*i}\right)^2}{8}\,. \tag{5.7.34}$$

In order to show our desired bound we also need to lower bound the value of $N_{ii^*}(t)$. Using Equation 5.7.6 we have that

$$R_{ii^*}(t) = M_{ii^*}(t) \cdot (P^t_{i^*|ii^*} - P^t_{i|ii^*}) \geq R_{\max} - 1 \implies M_{ii^*}(t) \geq \frac{R_{\max} - 1}{(P^t_{i^*|ii^*} - P^t_{i|ii^*})}$$

Using Lemma 5.7.3 we also know that

$$N_{ii^*}(t) \geq \frac{(P^t_{i^*|ii^*} + P^t_{i|ii^*})}{2} \cdot M_{ii^*}(t)$$

Combining this with the above we know that

$$N_{ii^*}(t) \geq (P^t_{i^*|ii^*} + P^t_{i|ii^*}) \cdot \frac{R_{\max} - 1}{2(P^t_{i^*|ii^*} - P^t_{i|ii^*})} = \frac{R_{\max} - 1}{2\Delta^t_{i^*i}} \geq \frac{8\log(nCT)}{\left(\Delta^t_{i^*i}\right)^2} \tag{5.7.35}$$

Combining Equations 5.7.34 and 5.7.35 we get the desired bound of Equation 5.7.33. Hence, for any $t' > t$ arm i will not be the part of the active set as $\mathcal{J}_i(t, C)$ will be 1. This implies that the regret cannot strictly exceed $R_{\max}$ leading to a contradiction.

Combining the bound for each term in Equation 5.7.28 we get that

$$\begin{aligned}\mathbf{E}\left[\sum_{t=1}^{T} r(S_t, i) \cdot \mathbb{1}[a_t = i^*, i \in S_t]\right] &\leq \frac{512\log(nCT)}{\Delta_{i^*i}} + 1 + \frac{12}{T} \cdot T \\ &\leq \frac{512\log(nCT)}{\Delta_{i^*i}} + 13\,.\end{aligned}$$

□

5.7.2.8 Proof of Lemma 5.7.8

Proof. For this lemma, we will assume without loss of generality, that arms are indexed in the order of decreasing weights, so that $w_1 > w_2 \geq w_3 \geq \cdots \geq w_n$, and $i^* = 1$. The proof of this lemma follows using a similar analysis as Yue et al. (2009) for the IF algorithm. The idea is to think of the sequence of anchor arms $a_1, a_2, a_3 \cdots$ as a random walk over a graph over n node where node $i \in [n]$ corresponds to the bandit arm $i \in [n]$. The probability of transition from node i to node j is the probability that WBA-L choses arm j as the next anchor when the current anchor is arm i. GCC node 1 is the absorbing node in the random walk, and the goal is to find the absorption time to node 1 in this random walk. Given that the execution of WBA-L is mistake-free and given the current anchor is j, the linear order of weights under the MNL model ensures that, for $i < i' < j$, the probability that arm i becomes the next anchor is greater than equal to the probability that arm i' becomes the next anchor, and these probabilities can be equal in the worst case. Also, given that the execution is mistake-free and given the current anchor is j, for $i'' \geq j$ the probability that arm i'' becomes the next anchor is 0. Hence, given that the random walk is at node j, it jumps to any $1, \cdots, j-1$ uniformly at random. Lemma 5 in Yue et al. (2009) shows that the probability that the random walk arrives at arm i is upper bounded $1/i$. Hence, the proof of this lemma follows from Lemma 5 in Yue et al. (2009). □

5.7.2.9 Proof of Lemma 5.7.9

Proof. We need to show that

$$\begin{aligned}\mathbf{E}\left[R_{ij}(T) \;\middle|\; \mathcal{E}_\lambda\right] \leq \Pr(\exists t \in [T] : a_t = j | \mathcal{E}_\lambda) \cdot \left(\frac{2048 \log(nCT)}{\Delta^{\mathrm{MNL}}_{i^* i}} + 1\right) \\ + \mathbf{E}\left[\sum_{t=1}^{T} \mathbb{1}[a_t = j, i \in S_t, i^* \notin A_t] \;\middle|\; \mathcal{E}_\lambda\right].\end{aligned}$$

In order to show the above bound we will consider three cases:

Case 1: $w_i < w_j$ and $\Delta^{\text{MNL}}_{i^*j} \le \Delta^{\text{MNL}}_{ji}$. In this case we have that $w_{i^*} - w_j \le w_j - w_i$, hence, we will use the first condition in $\mathcal{E}_\lambda$ to show that

$$\left| \frac{\widehat{P}^t_{i|ji}}{\widehat{P}^t_{i|ji} + \widehat{P}^t_{j|ji}} - \frac{w_i}{w_i + w_j} \right| \le \sqrt{\frac{4\Delta^{\text{MNL}}_{ji}\lambda}{R_{ij}(t)}} + \frac{4\Delta^{\text{MNL}}_{ji}\lambda}{3R_{ij}(t)} .$$

Similar to the proof of Lemma 5.7.6 we define $R_{\max} := \frac{1024\log(nCT)}{\Delta^{\text{MNL}}_{i^*i}} + 1$ and show that the regret in this case is upper bounded by $R_{\max}$. We will again prove this by contradiction similar to the proof of Lemma 5.7.6, by assuming that there is a trail t such that the regret exceeds $R_{\max}$. We will then have that

$$\sqrt{\frac{4\Delta^{\text{MNL}}_{ji}\lambda}{R_{\max} - 1}} + \frac{4\Delta^{\text{MNL}}_{ji}\lambda}{3(R_{\max} - 1)} \le \sqrt{\frac{\Delta^{\text{MNL}}_{ji} \cdot \Delta^{\text{MNL}}_{i^*i}}{64}} + \frac{\Delta^{\text{MNL}}_{ji} \cdot \Delta^{\text{MNL}}_{i^*i}}{192} \le \frac{\Delta^{\text{MNL}}_{ji}}{4} ,$$

where the above inequality follows from the fact that $\Delta^{\text{MNL}}_{i^*i} \le 2\Delta^{\text{MNL}}_{ji}$. Given the event $\mathcal{E}_\lambda$, we have that

$$\left| \frac{\widehat{P}^t_{i|ji}}{\widehat{P}^t_{i|ji} + \widehat{P}^t_{j|ji}} - \frac{w_i}{w_i + w_j} \right| \le \frac{\Delta^{\text{MNL}}_{ji}}{4} .$$

We also have that

$$\begin{aligned} 2M_{ij}(t) \cdot (P^t_{j|ij} - P^t_{i|ij}) &= \sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \subseteq S_{t'}] \cdot 2 \cdot \frac{w_j - w_i}{\sum_{a \in S_{t'}} w_a} \\ &\ge \sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \subseteq S_{t'}] \cdot \frac{w_{i^*} - w_i}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a} \\ &= R_{ij}(t) \ge R_{\max} - 1 \end{aligned}$$

Now, combined with Lemma 5.7.3 to show that $N_{ij}(t) \ge 8\log(nCT)/(\Delta^{\text{MNL}}_{ji})^2$, and following along an argument similar to Lemma 5.7.6, we can show that arm i will be eliminated on or before trial t. Therefore, the regret cannot strictly exceed $R_{\max}$ which is a contradiction. Moreover, the regret in this case is 0 if arm j does not become the anchor. Hence, the final regret in this case is upper bounded by $P(\exists t \in [T] : a_t = j | \mathcal{E}_\lambda) R_{\max}$.

Case 2: $w_i < w_j$ and $\Delta^{\text{MNL}}_{i^*j} > \Delta^{\text{MNL}}_{ji}$. This condition implies that $w_{i^*} - w_j > w_j - w_i$. In this case we will bound the regret of arm i using the regret of arm j.

$$\begin{aligned}
\mathbf{E}\left[R_{ij}(t)|\mathcal{E}_\lambda\right] &= \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \in S_{t'}] \left(\frac{w_{i^*} - w_i}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&\leq \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \in S_{t'}] \left(\frac{2(w_{i^*} - w_j)}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&\leq \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i^*,j\} \in S_{t'}] \left(\frac{2(w_{i^*} - w_j)}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&= \mathbf{E}\left[2R_{j|i^*}(t)|\mathcal{E}_\lambda\right] \ ,
\end{aligned}$$

where the second last inequality follows from the fact that the partition of arms is random in nature, hence the expected weight of set $S_{t'} \ni i$ is that same as the expected weight of set $S_{t'} \ni i^*$. We will now use the second condition in $\mathcal{E}_\lambda$ to show that

$$\left|\frac{\widehat{P}^t_{j|i^*j}}{\widehat{P}^t_{j|i^*j} + \widehat{P}^t_{i^*|i^*j}} - \frac{w_j}{w_j + w_{i^*}}\right| \leq \sqrt{\frac{2\Delta^{\text{MNL}}_{i^*j}\lambda}{R_{j|i^*}(t)}} + \frac{2\Delta^{\text{MNL}}_{i^*j}\lambda}{3R_{j|i^*}(t)} \, .$$

Similar to the proof of Lemma 5.7.6 we define $R_{\max} := \frac{512\log(nCT)}{\Delta^{\text{MNL}}_{i^*j}} + 1 \leq \frac{1024\log(nCT)}{\Delta^{\text{MNL}}_{i^*i}} + 1$ and show that the regret $R_{j|i^*}$ in this case is upper bounded by $R_{\max}$. We will again prove this by contradiction similar to the proof of Lemma 5.7.6, by assuming that there is a trail t such that the regret exceeds $R_{\max}$. We will then have that

$$\sqrt{\frac{2\Delta^{\text{MNL}}_{i^*j}\lambda}{R_{\max} - 1}} + \frac{2\Delta^{\text{MNL}}_{i^*j}\lambda}{3(R_{\max} - 1)} \leq \sqrt{\frac{\Delta^{\text{MNL}}_{i^*j} \cdot \Delta^{\text{MNL}}_{i^*j}}{32}} + \frac{\Delta^{\text{MNL}}_{i^*j} \cdot \Delta^{\text{MNL}}_{i^*j}}{96} \leq \frac{\Delta^{\text{MNL}}_{i^*j}}{4} \, ,$$

where the above inequality follows from the fact that $\Delta^{\text{MNL}}_{i^*i} \leq 2\Delta^{\text{MNL}}_{ji}$. Given the event $\mathcal{E}_\lambda$, we have that

$$\left|\frac{\widehat{P}^t_{j|i^*j}}{\widehat{P}^t_{j|i^*j} + \widehat{P}^t_{i^*|i^*j}} - \frac{w_j}{w_j + w_{i^*}}\right| \leq \frac{\Delta^{\text{MNL}}_{i^*j}}{4} \, .$$

We also have that $R_{j|i^*}(t) = M_{i^*j}(t) \cdot (P^t_{i^*|i^*j} - P^t_{j|i^*j})$. Now, combined with Lemma 5.7.3 to show that $N_{i^*i}(t) \geq 8\log(nCT)/(\Delta^{\text{MNL}}_{i^*i})^2$, and following along an argument similar to

Lemma 5.7.6, we can show that arm j will be replaced by a new anchor. Therefore, the regret $R_{i|i^*}$ cannot strictly exceed $R_{\max}$ which is a contradiction. Moreover, the regret in this case is 0 if arm j does not become the anchor. Hence, the final regret in this case is upper bounded by $P(\exists t \in [T] : a_t = j|\mathcal{E}_\lambda)R_{\max}$.

Case 3: $w_i \geq w_j$. In this case again we will bound the regret of arm i using the regret of arm j.

$$\begin{aligned}
\mathbf{E}\left[R_{ij}(t)|\mathcal{E}_\lambda\right] &= \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \in S_{t'}] \left(\frac{w_{i^*} - w_i}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&\leq \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i,j\} \in S_{t'}] \left(\frac{w_{i^*} - w_j}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&\leq \mathbf{E}\left[\sum_{t'=1}^{t} \mathbb{1}[a_{t'} = j, \{i^*,j\} \in S_{t'}] \left(\frac{w_{i^*} - w_j}{\sum_{a \in S_{t'} \cup \{i^*\}} w_a}\right) \Big| \mathcal{E}_\lambda\right] \\
&= \mathbf{E}\left[R_{j|i^*}(t)|\mathcal{E}_\lambda\right] ,
\end{aligned}$$

where the second last inequality follows from the fact that the partition of arms is random in nature, hence the expected weight of set $S_{t'} \ni i$ is that same as the expected weight of set $S_{t'} \ni i^*$. Using the argument in case 2, one can show that the final regret in this case is upper bounded by $P(\exists t \in [T] : a_t = j|\mathcal{E}_\lambda)R_{\max}$. Finally, the regret due to anchor can be bounded by the regret of any other arm that is player with the anchor. □

5.8 Conclusion

We have introduced a new framework for bandit learning from choice feedback that generalizes the dueling bandit framework. Our main result is to show that computationally efficient learning is possible in this more general framework under a wide class of choice models that is considerably more general than the previously studied class of MNL models. Our algorithms for this general setting, achieve (almost) optimal regret for the GCC class of models. For the special case $k = 2$, our algorithms are competitive with previous dueling bandit algorithms;

for $k > 2$, our algorithms outperform the recently proposed MaxMinUCB (MMU) algorithm even on MNL models for which MMU was designed.

Chapter 6

Finding the Best Coin with Limited Adaptivity

In this chapter we start our discussion at the interface of machine learning and theoretical computer science. We will study how to find the most biased coin from a set of coins using parallel interactions– a problem that has applications in both machine learning and theoretical computer science.

6.1 Introduction

6.1.1 Background

In the classical machine learning settings, the learner is a passive observer who is given a collection of randomly sampled observations from which to learn. In recent years, there has been growing interest in *active learning* models, where the learner can actively request labels or feedback at specific data points; the hope is that, by adaptively guiding the data collection process, learning can be accomplished with fewer observations than in the passive case. Most learning algorithms operate in one of these settings: learning is either fully passive, or fully active.

In an increasing number of applications, while active querying is possible, the number of *rounds* of interaction with the feedback generation mechanism is limited. For example, in crowdsourcing, one can actively request feedback by sending queries to the crowd, but there is typically a waiting time before queries are answered; if the overall task is to be completed within a certain time frame, this effectively limits the number of rounds of interaction. Similarly, in marketing applications, one can actively request feedback by sending surveys to customers, but there is typically a waiting time before survey responses are received; again, if the marketing campaign is to be completed within a certain time frame, this effectively limits the number of rounds of interaction.

In this chapter, we study active/adaptive learning with *limited rounds of adaptivity*, where the learner can actively request feedback at specific data points, but can do so in only a small number of rounds. Specifically, the learner is free to query any number of data points in each round; however, all data points to be queried in a given round must be submitted *simultaneously*, based only on feedback received in previous rounds. In this setting, we are interested not only in bounding the overall query complexity of the learner, but rather in understanding the tradeoff between the number of rounds and the overall query complexity: how many queries are needed given a fixed number of rounds?

We study this question in the context of an abstract coin tossing problem, and discuss how the results give us novel insights into the round vs. query complexity tradeoff for two problems that have received increasing interest in the learning theory community in recent years: multi-armed bandits, and ranking from pairwise comparisons[1].

The abstract coin problem we study can be described as follows: say we are given n coins with unknown biases, each of which can be 'queried' by tossing the coin and observing the outcome of the toss. The goal is to find the k coins with highest biases. This problem is a special case of the problem of finding the k best arms in a stochastic multi-armed bandit (MAB), and has received considerable attention in recent years (Even-Dar et al., 2006; Kalyanakrishnan and Stone, 2010; Audibert and Bubeck, 2010; Kalyanakrishnan et al., 2012; Gabillon et al., 2012; Jamieson et al., 2013; Bubeck et al., 2013; Karnin et al., 2013; Chen and Li, 2015; Kaufmann et al., 2016; Jun et al., 2016; Chen et al., 2017a). In particular, it is known that $O\big(\frac{n\log k}{\Delta_k^2}\big)$ coin tosses suffice to find the k most biased coins with arbitrarily high constant probability, where Δ_k is the gap between the k-th and $(k+1)$-th largest biases (Kalyanakrishnan and Stone, 2010; Even-Dar et al., 2006). It is also known that this bound is optimal in terms of the worst-case query complexity (Kalyanakrishnan et al., 2012; Mannor and Tsitsiklis, 2004). (see Table 4; see also Section 6.2 for the exact definition of parameters involved). However, the previous best algorithms for this problem all required $\Omega(\log n)$ rounds of adaptivity to

[1] In the MAB and ranking literature, the query complexity of an algorithm is often referred to as simply its *sample complexity*. In this chapter we use the two terms interchangeably.

achieve the optimal worst-case query complexity. But are $\Omega(\log n)$ rounds necessary for achieving this optimal query complexity? In this chapter we seek to answer this question by designing an algorithm that requires much less that $\log n$ rounds of adaptivity.

6.1.2 Our Contributions

We present an algorithm, Agressive-Elimination, that significantly improves upon the round complexity of state-of-the-art algorithms, yet still achieves the optimal worst-case query complexity: given the gap parameter Δ_k, our algorithm returns the k most biased coins using $O\big(\frac{n\log k}{\Delta_k^2}\big)$ coin tosses with arbitrarily large constant probability in only $\log^*(n)$ rounds of adaptivity. The algorithm proceeds in rounds and in each round performs: *(i)* an "estimation" phase to approximate the bias of each coin, and *(ii)* an "elimination" phase to reduce the number of possible candidates and finds the top k most biased coins among the remaining candidates in the subsequent rounds. The elimination phase gets more "aggressive" over the rounds: in each round, the number of remaining coins reduces to an exponentially smaller fraction (across different rounds) of the current coins. This allows the algorithm to find the top k most biased coins in only $\log^* n$ rounds of adaptivity (as opposed to $\log n$ if the fraction was constant throughout). Figure 14 gives an example of the rate at which items are eliminated per round for Agressive-Elimination algorithm, and the $\log n$-round Halving algorithm (Kalyanakrishnan and Stone, 2010; Even-Dar et al., 2006). The main insight behind our algorithm is that by removing more and more coins in the elimination phase we can allocate more and more budget (i.e., samples for each remaining coin) to the estimation phase which in turn results in even more decrease in the number of candidate coins for the next round.

Finally, we address the question of round vs. query complexity tradeoff for this problem in a more fine-grained level: For any fixed number of rounds r, we present an algorithm for the above coin problem that uses $O\big(\frac{n}{\Delta_k^2}(\text{ilog}^{(r)}(n) + \log k)\big)$ coin tosses. Here, $\text{ilog}^{(r)}(\cdot)$ denotes the iterated logarithm of order r. Our results provide a near-complete understanding of the power of each additional round of adaptivity in reducing the query complexity of the

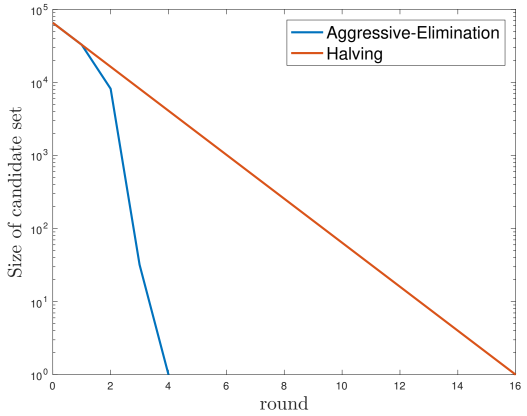


Figure 14: An example illustrating that our algorithm eliminates items more "aggresively" as compared to the Halving algorithm of Kalyanakrishnan and Stone (2010); Even-Dar et al. (2006). Here, $n = 2^{16}$ and $k = 1$.

Table 4: Summary of some results for k best arms identification in stochastic multi-armed bandits.

	Algorithm	**# Rounds of Adaptivity**	**Sample/Query Complexity**
$k = 1$	Even-Dar et al. (2002)	$\Theta(\log(n))$	$O(\frac{n\log(1/\delta)}{\Delta_1^2})$
	Audibert and Bubeck (2010)	$\Theta(n)$	$O\left(\sum_{i=1}^{n}\Delta_i^{-2}\cdot\log^2(\frac{n}{\delta})\right)$
	Chen and Li (2015)	$\Omega(\log(n))$	$O\left(\sum_{i=1}^{n}\Delta_i^{-2}\cdot\log(\frac{\log(\min\{n,\Delta_i^{-1}\})}{\delta})\right)$
All $k \in [n]$	Kalyanakrishnan and Stone (2010)	$\Theta(\log(n))$	$O(\frac{n\log(k/\delta)}{\Delta_k^2})$
	Bubeck et al. (2013)	$\Theta(n)$	$O\left(\sum_{i=1}^{n}\Delta_i^{-2}\cdot\log^2(\frac{n}{\delta})\right)$
	This work	$\log^*(n)$	$O(\frac{n\log(k/\delta)}{\Delta_k^2})$

algorithms for this problem.

Our results for the above coin problem are also applicable to the problem of top-k ranking from pairwise comparisons, another problem that has received considerable interest in recent years (Feige et al., 1994; Busa-Fekete et al., 2013; Chen and Suh, 2015; Shah and Wainwright, 2015; Jang et al., 2016; Heckel et al., 2016; Davidson et al., 2014; Braverman et al., 2016a). Most top-k ranking approaches we are aware of assume either a non-adaptive setting or a fully adaptive setting; the main exceptions to this are Feige et al. (1994); Davidson et al. (2014); Braverman et al. (2016a), who consider the top-k ranking problem under limited rounds of adaptivity, but under the restricted *noisy permutation* model of pairwise comparisons

Table 5: Summary of some results on top-k ranking from pairwise comparisons.

	Pairwise Comparison Model	**# Rounds of Adaptivity**	**Sample/Query Complexity**
Chen and Suh (2015)	Bradley-Terry-Luce	Non-adaptive	$O\left(\frac{n\log(n/\delta)}{(w_{[k]}-w_{[k+1]})^2}\right)$
Shah and Wainwright (2015)	General	Non-adaptive	$O(\frac{n\log(n/\delta)}{\Delta_k^2})$
Braverman et al. (2016a)	Noisy Permutation	4	$O\left(\frac{n\log(n/\delta)}{(1-2p)^2}\right)$
Busa-Fekete et al. (2013), Heckel et al. (2016)	General	$\Omega\left(\Delta_k^{-2}\cdot\log(n)\right)$	$O\left(\sum_{i=1}^n \Delta_i^{-2}\cdot\log(\frac{n}{\delta\Delta_i})\right)$
This work	General	$\log^*(n)$	$O(\frac{n\log(k/\delta)}{\Delta_k^2})$

(defined in Section 6.4). In our work, we make no assumptions on the underlying pairwise comparison model. Again, our results for the abstract coin problem above give us a novel algorithm for top-k ranking from pairwise comparisons that requires only $\log^*(n)$ rounds; to our knowledge, this is the first study of this problem under general pairwise comparison models in the limited-adaptivity setting. See Table 5 for a summary (see also Section 6.4 for the exact definition of parameters involved).

Our work shows that for a well-studied class of learning problems, the power of fully adaptive exploration in minimizing worst-case query complexity is realizable by just a few rounds of adaptive exploration. In fact, for any realistic input size for the problems considered here, our work shows that at most 5 adaptive rounds are needed to realize optimal worst-case query complexity. We hope that our techniques can be used for other classes of learning problems to gain an insight into how the query complexity changes as one interpolates between the fully passive and fully active settings.

Remark 6.1.1. *Agarwal et al. (2017a) also shows that* $\log^*(n)$ *rounds are necessary for any algorithm that achieves the optimal query complexity bound. This lower bound result is a contribution of Sepehr Assadi's thesis who was a co-author in this work.*

6.1.3 Related Work

The general question of computation with limited rounds of adaptivity has been studied for certain problems such as sorting and selection in the theoretical computer science (TCS) literature under the term *parallel algorithms* (Valiant, 1975; Bollobás and Thomason, 1983; Ajtai et al., 1986; Pippenger, 1987; Alon and Azar, 1988; Cole, 1988; Bollobás and Brightwell, 1990; Feige et al., 1994; Davidson et al., 2014; Braverman et al., 2016a). However, with the exception of Feige et al. (1994); Davidson et al. (2014); Braverman et al. (2016a), these studies all operate in a deterministic setting, where any sample yields a deterministic outcome; this is unlike the setting we consider in our problems, where there is an underlying probabilistic model and queries yield noisy outcomes.

We note that the coin problem studied by Karp and Kleinberg (2007) is different from ours: there, given a ranked list of coins with unknown biases and a target bias $p \in (0, 1)$, the goal is to find the coins that have bias greater than p. In our case we do not know a ranking on the coins. Another line of work on biased coin identification is that of Chandrasekaran and Karp (2014); Malloy et al. (2012); Jamieson et al. (2016): there, given an infinite population of coins, each of which is of one of two types, 'heavy' or 'light', the goal is to identify a coin of the heavy type. In our case we have a finite population of coins, each of which can be of a different type. Moreover, all these previous papers work in the fully adaptive setting, while our focus is on the limited-adaptivity setting.

The problem of best arm identification in MABs has mostly been considered in a fully adaptive setting, where the learner can observe the outcome of any arm pull before selecting the next arm to be pulled (Even-Dar et al., 2006; Audibert and Bubeck, 2010; Kalyanakrishnan et al., 2012; Gabillon et al., 2012; Jamieson et al., 2013; Bubeck et al., 2013; Karnin et al., 2013; Hillel et al., 2013; Perchet et al., 2015b; Chen and Li, 2015; Kaufmann et al., 2016; Jun et al., 2016; Chen et al., 2017a). A recent work by Jun et al. (2016) is most closely related to our work. It considers algorithms that pull multiple arms in each round and there is a bound on the number of arms that the algorithm is allowed to pull in each round. However, the number of rounds required by their algorithm in the worst-case is $\Omega(\log(n))$ irrespective of the bound on the number of pulls in each round.

The problem of top-k ranking from (noisy) pairwise comparisons has mostly been considered in either the non-adaptive setting or the fully adaptive setting (Busa-Fekete et al., 2013; Chen and Suh, 2015; Shah and Wainwright, 2015; Jang et al., 2016; Heckel et al., 2016). Feige et al. (1994), and more recently Davidson et al. (2014); Braverman et al. (2016a), considered a setting with limited rounds of adaptivity, but under a restricted pairwise comparison model that we refer to as the *noisy permutation* model (see Section 6.4 for details). In contrast, in this work, we make no assumptions on the underlying pairwise comparison model.

6.1.4 Notation

For any integer $a \geq 1$, $[a] := \{1, \ldots, a\}$. For a (multi-)set of numbers $\{a_1, \ldots, a_n\}$, we define $a_{[i]}$ as the i-th largest value in this set (ties are broken arbitrarily). For any integer $r \geq 0$, $\text{ilog}^{(r)}(a)$ denotes the iterated logarithms of order r, i.e. $\text{ilog}^{(r)}(a) = \max\left\{\log\left(\text{ilog}^{(r-1)}(a)\right), 1\right\}$ and $\text{ilog}^{(0)}(a) = a$. Matrices and vectors are denoted in boldface, e.g., $\boldsymbol{A}$ and $\boldsymbol{b}$, and random variables in serif, e.g., X.

6.1.5 Organization

We start by formalizing the coin tossing abstraction we use in this paper in Section 6.2. Section 6.3 presents our algorithm. In Section 6.4, we present our results for the ranking problem as a corollary of the results for the most biased coins problem. We present an extension of our results to the case of sub-Gaussian rewards in Section 6.5. We conclude in Section 6.6.

6.2 Finding the k Most Biased Coins / k Best Arms

Here, we present our main results on finding the k most biased coins using coin tosses with a limited number of rounds of adaptivity. We give an algorithm for this problem in Section 6.3 that achieves an optimal worst-case tradeoff between round and query complexity. The coin problem is equivalent to the problem of the k best arms identification problem in MABs with Bernoulli reward distributions. Our results also extend to the more general case of MABs with sub-Guassian reward distributions (see Section 6.5).

The specific problem we consider can be stated formally as follows: given n coins with unknown biases $p_1, \ldots, p_n$, and an integer $k \in [n]$, the goal is to identify (via tosses of the n coins) the set of k most biased coins. An important parameter in determining the query complexity of this problem is the *gap parameter* $\Delta_k := p_{[k]} - p_{[k+1]}$, i.e. the gap between the k-th and $(k+1)$-th highest biases (recall that $p_{[i]}$ denotes the bias of the i-th most biased coin). We also define $\Delta_i = \max\{\left|p_{[i]} - p_{[k+1]}\right|, \left|p_{[i]} - p_{[k]}\right|\}$. We will assume throughout that the set of k most biased coins is unique, i.e. that $\Delta_k > 0$; we will also assume our algorithm is given a lower bound Δ on the gap parameter ($\Delta_k \geq \Delta > 0$).[2]

We are interested here in algorithms that require limited *rounds* of adaptivity. In each round, an algorithm can decide to query various coins by tossing them (with no limit on the number of coins that can be tossed in a round or on the number of times any given coin can be tossed in a round); however, all tosses to be conducted in a given round must be chosen *simultaneously*, based only on the outcomes observed in previous rounds. We say an algorithm is an *r-round algorithm* if it uses at most r rounds of adaptivity; the total number of coin tosses it uses is termed its *query complexity*. For any $\delta \in [0, 1)$, we say an algorithm is a *δ-error algorithm* for the above problem if it correctly returns the set of k most biased coins with probability at least $1 - \delta$.

6.3 A Limited-Adaptivity Algorithm for Finding the k Most Biased Coins

Our main algorithmic result is the following:

Theorem 6.3.1. *There exists an algorithm that given an integer $k \in [n]$, a set of n coins with gap parameter $\Delta_k \in (0, 1)$, target number of rounds $r \geq 1$, and confidence parameter $\delta \in [0, 1)$, finds the set of k most biased coins w.p. $\geq 1 - \delta$ using $O\left(\frac{n}{\Delta_k^2} \cdot \left(\text{ilog}^{(r)}(n) + \log(k/\delta)\right)\right)$ coin tosses and r rounds of adaptivity.*

[2] We point out that the assumption that $\Delta_k > 0$ is *only* for simplicity of exposition; by picking Δ_k to be the gap between the bias of the k-th most biased coin and the next largest *distinct* bias value, our algorithm works as it is. The assumption about knowledge of Δ is also common in the MAB and ranking literature; see, e.g., (Even-Dar et al., 2006; Kalyanakrishnan and Stone, 2010; Chen and Suh, 2015; Shah and Wainwright, 2015).

We also point out that by setting $r = \log^*(n)$ in Theorem 6.3.1, we can achieve the *optimal worst-case query complexity* (Kalyanakrishnan et al., 2012; Mannor and Tsitsiklis, 2004) in a significantly smaller number of rounds of adaptivity than previous work.

Corollary 6.3.2. *There exists an algorithm that given an integer $k \in [n]$, a set of n coins with gap parameter $\Delta_k \in (0,1)$, and confidence parameter $\delta \in [0,1)$, finds the set of k most biased coins w.p. $\geq 1-\delta$ using $O\left(\frac{n}{\Delta_k^2} \cdot \log(k/\delta)\right)$ coin tosses and only $\log^*(n)$ rounds of adaptivity.*

6.3.1 Algorithm

We design a recursive algorithm, which we term as Agressive-Elimination, for proving Theorem 6.3.1. The pseudo-code is given in Algorithm 9. It takes as input a set $S \subseteq [n]$ of $m \geq k$ *candidate* coins for the top k coins and a parameter r denoting the number of rounds of adaptivity the algorithm can use. In addition, the algorithm is given the confidence parameter $\delta \in (0,1)$ and a lower bound on the gap parameter $\Delta \leq \Delta_k$. Given this input, Algorithm 9 essentially does the following:

1. **Estimation phase:** Toss each coin $O\left(\frac{1}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log(k/\delta)\right)\right)$ many times and estimate the bias of each coin.
2. **Elimination phase:** Let S' be the set of $O(\frac{m}{\text{ilog}^{(r-1)}(m)})$ coins with the largest estimated biases. Recursively solve the problem for the set S' in the remaining $r-1$ rounds.

We point out that the estimation phase of the algorithm is allowed to be erroneous, i.e. there might be large deviations between the estimated biases and the true biases for a relatively large fraction of coins. The elimination phase is then designed to be robust to such errors by selecting a suitably large subset for the next round. As rounds progress, the set of candidates for k most biased coins shrinks more and more such that in the last round, the algorithm can estimate the bias of each candidate with high confidence and return the k most biased coins. We should also point that in any round, if the input set S becomes too small, i.e. is of

Algorithm 9 Agressive-Elimination$(S_r, k, r, \delta, \Delta)$

```
Input: set $S_r \subseteq [n]$ of coins, number of desired top items $k$, number of rounds $r$, confidence parameter $\delta \in (0,1)$, and lower bound on gap parameter $\Delta \leq \Delta_k$
Let $m = m_r = |S_r|$ and $t_r := \frac{2}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log(8k/\delta)\right)$.
Toss each coin $i \in S_r$ for $t_r$ times.
For each $i \in S_r$, define $\widehat{p}_i$ as the fraction of times coin $i$ turns up heads.
Sort the coins in $S_r$ in a decreasing order of $\widehat{p}$-values.
if $r = 1$ then
    Return: the set of $k$ most biased coins (according to $\widehat{p}$-values).
else
    Let $m_{r-1} := k + \frac{m}{\text{ilog}^{(r-1)}(m)}$ and $S_{r-1}$ be the set of $m_{r-1}$ most biased coins according to $\widehat{p}$.
end if
if $m_{r-1} \leq 2k$ then
    Return: Agressive-Elimination$(S_{r-1}, k, 1, \delta/2, \Delta)$.
else
    Return: Agressive-Elimination$(S_{r-1}, k, r-1, \delta/2, \Delta)$.
end if
```

size $O(k)$, then Algorithm 9 bypasses the subsequent rounds and simply runs the 1-round algorithm on this set to recover the answer.

6.3.2 Analysis

We present the proof of Theorem 6.3.1 in detail in this section. Throughout this section, for any algorithm $\mathcal{A}$, $\mathsf{cost}(\mathcal{A})$ denotes the query complexity of $\mathcal{A}$ and $\mathsf{deg}(\mathcal{A})$ denotes the degree of adaptivity it uses, i.e., its round complexity. We start by providing a high level overview of the proof.

Overview: To illustrate the main ideas behind our algorithm, we focus on the case that $k = 1$. Consider the following type of input for best k coins problem: there exists a single *heavy* coin and $n-1$ *light* coins with the gap of Δ between the bias of the heavy coin and any light coin. It follows from a simple application of the Hoeffding's bound that for any $\delta \in (0,1)$, $O(\log(1/\delta)/\Delta^2)$ coin tosses are sufficient to distinguish whether a single coin is heavy or not with probability $1-\delta$. We can now use this simple observation to design an r-round algorithm for each number of rounds r.

The case of $r = 1$ is quite simple: simply set $\delta = \Theta(\frac{1}{n})$ and a union bound ensures that with some constant probability, *every* coin is distinguished correctly, which allows us to output the heavy coin correctly. Now consider the case when $r = 2$. Here, the limited budget for 2-round algorithms in Theorem 6.3.1 does not allow us to distinguish every coin correctly in the first round of coin tossing. Instead, we make the following simple yet crucial observation: it is enough for us to only classify the heavy coin and a large fraction of light coins correctly in the first round. Indeed by setting the parameter $\delta = \Theta(\frac{1}{\log n})$ (i.e., performing $O(n \log\log n/\Delta^2)$ coin tosses in the first round), we can reduce the set of possible choices for the heavy coin to roughly $n/\log n$ coins. But then our budget allows us to run the previous 1-round algorithm in the second round on this *smaller* set of coins to find the heavy coin. This results in the total number of coins tosses being $O(n \log\log n/\Delta^2)$ (in the first round) plus $O\left((n/\log n) \cdot \log\left(n/\log n\right)/\Delta^2\right) = O(n/\Delta^2)$ (in the second run), which matches the bounds for the $r = 2$ case in Theorem 6.3.1.

This discussion leads us to the following generic r-round algorithm: perform a number of coin tosses in the first round to recover a sufficiently smaller set that almost surely contains the heavy coin; recursively solve the problem on the remaining coins using the $(r-1)$-round version of the algorithm in the subsequent rounds. Here, "sufficiently smaller set" should be chosen such that the query complexity of an $(r-1)$-round algorithm on this set is within the budget of the r-round algorithm (over the original set of coins). Exploiting this approach to its fullest allows us to design our r-round algorithm for any number of rounds r and prove Theorem 6.3.1.

We now provide a more formal proof for the theorem by proving Lemma 6.3.3 and then providing a bound on the number of coin tosses that our algorithm makes in Lemma 6.3.5.

Lemma 6.3.3. *Suppose S is any subset of coins $[n]$ with size m and gap parameter $\Delta \leq \Delta_k$ such that $[k] \subseteq S$. For any number of rounds $1 \leq r \leq \log^*(m) - 3$ and any confidence parameter $\delta \in (0, 1)$, Algorithm 9 returns the set of k most biased coins w.p. at least $1 - \delta$.*

Before proving Lemma 6.3.3, we need the following simple claim. In the remainder of this section, we fix $\varepsilon := \Delta/2$.

Claim 6.3.4. *For any round $r \geq 1$, and any coin $i \in S_r$,*

$$\Pr\left(|\widehat{p}_i - p_i| \geq \varepsilon\right) \leq \frac{\delta}{4k \cdot \mathrm{ilog}^{(r-1)}(m)}.$$

Proof. By Hoeffding's inequality, we have,

$$\begin{aligned}\Pr\left(|\widehat{p}_i - p_i| \geq \varepsilon\right) &\leq 2\exp\left(-2\epsilon^2 \cdot t_r\right)\\ &\leq 2\exp\left(-\left(\mathrm{ilog}^{(r)}(m) + \log(8k/\delta)\right)\right) \leq \frac{\delta}{4k \cdot \mathrm{ilog}^{(r-1)}(m)}\end{aligned}$$

as $\mathrm{ilog}^{(r)}(m) = \log \mathrm{ilog}^{(r-1)}(m)$. □

In the following, for any integer $r \geq 1$, we use $\mathcal{A}_r$ to denote Algorithm 9 with r number of rounds. We now prove Lemma 6.3.3.

Proof. (**of Lemma 6.3.3.**)

The proof is by induction on the number of rounds r.

Base case: The base case follows immediately from Claim 6.3.4. Indeed for $r = 1$, Claim 6.3.4 ensures that for any $i \in S_1$,

$$\Pr\left(|\widehat{p}_i - p_i| \geq \varepsilon\right) \leq \frac{\delta}{4k \cdot \mathrm{ilog}^{(0)}(m_1)} \leq \frac{\delta}{m_1}$$

as $\mathrm{ilog}^{(r-1)}(m_1) = m_1$ by definition. By taking a union bound over all m_1 coins, we obtain that w.p. $1 - \delta$, simultaneously for all coins $i \in S_1$, $|\widehat{p}_i - p_i| < \varepsilon$. This implies that w.p.

$1 - \delta$,

$$\begin{aligned} &\forall i \in [k] && \widehat{p}_i > p_i - \varepsilon = p_i - \Delta/2 \geq p_k - \Delta/2 \\ &\forall j \in S_1 \setminus [k] && \widehat{p}_j < p_j + \varepsilon \leq p_j + \Delta/2 \leq p_{k+1} + \Delta/2 \end{aligned}$$

As $\Delta \leq p_k - p_{k+1}$, we obtain that the returned set of k most biased coins according to $\widehat{p}$-values is the correct answer, finalizing the proof of the base case.

Induction step: Suppose the lemma is true for all number of rounds smaller than $r \leq \log^*(m) - 3$ and we prove it for the case of r rounds, i.e., for $\mathcal{A}_r$. In particular, we need to show that $\mathcal{A}_r$ returns the set of k most biased coins with probability at least $1 - \delta$.

Let $I = \{i \in [k] : \widehat{p}_i < p_i - \varepsilon\}$ and $J = \{j \in S_r \setminus [k] : \widehat{p}_j > p_j + \epsilon\}$. We know that for all $i \in [k]$ and $j \in S_r \setminus [k]$, $p_i - p_j \geq 2\varepsilon$. As the algorithm identifies a set of $m_{r-1} = k + \frac{m_r}{\text{ilog}^{(r-1)}(m_r)}$ coins with the highest estimated biases (according to $\widehat{p}$) to recurse upon, we have,

$$\Pr(\mathcal{A}_r \text{ errs}) \leq \Pr(|I| > 0) + \Pr\left(|J| > \frac{m_r}{\text{ilog}^{(r-1)}(m_r)}\right) + \Pr(\mathcal{A}_{r-1} \text{ errs} \mid \mathcal{E}) \tag{6.3.1}$$

where $\mathcal{E}$ denotes the event that $|I| = 0$ and $|J| \leq \frac{m_r}{\text{ilog}^{(r-1)}(m_r)}$, i.e., the complement of the first two events above.

In the following, we bound probability of each event above. We first have,

$$\Pr(|I| > 0) \leq \sum_{i \in [k]} \Pr(\widehat{p}_i < p_i - \varepsilon) \leq_{\text{Claim 6.3.4}} k \cdot \frac{\delta}{4k \cdot \text{ilog}^{(r-1)}(m_r)} \leq \frac{\delta}{4} \tag{6.3.2}$$

where the last inequality is true because $\text{ilog}^{(r-1)}(m_r) \geq 1$.

We next bound the probability that $|J| > \frac{m_r}{\text{ilog}^{(r-1)}(m_r)}$. For all $j \in S_r \setminus [k]$, we define an indicator random variable Y_j which is 1 iff $\widehat{p}_j > p_j + \varepsilon$. We further define $\mathsf{Y} := \sum_j \mathsf{Y}_j$. We have,

$$\mathbb{E}\left[\mathsf{Y}\right] = \sum_j \mathbb{E}\left[\mathsf{Y}_j\right] = \sum_j \Pr\left(\widehat{p}_j > p_j + \varepsilon\right) \leq_{\text{Claim 6.3.4}} \sum_j \frac{\delta}{4k \cdot \text{ilog}^{(r-1)}(m_r)} \leq \frac{\delta \cdot m_r}{4 \cdot \text{ilog}^{(r-1)}(m_r)}$$

Notice that $\mathsf{Y} = |J|$; hence,

$$\Pr\left(|J| > \frac{m_r}{\text{ilog}^{(r-1)}(m_r)}\right) \leq \Pr\left(\mathsf{Y} > \frac{4}{\delta} \cdot \mathbb{E}\left[\mathsf{Y}\right]\right) \leq \frac{\delta}{4} \tag{6.3.3}$$

where the last inequality is by Markov bound.

Finally, we calculate the probability of error of $\mathcal{A}_{r-1}$ conditioned on that none of the two events above happens (i.e., the event $\mathcal{E}$). In that case, we have $[k] \subseteq S_{r-1}$ and that $\Delta \leq \Delta_k$. As $r \leq \log^*(m_r) - 3$ (by the lemma statement), we have $r - 1 \leq (\log^*(m_r) - 1) - 3 \leq \log^*(\log m_r) - 3 \leq \log^*(m_{r-1}) - 3$. Therefore, the input to $\mathcal{A}_{r-1}$ satisfies the assumptions in the lemma statement as well and since the confidence parameter for $\mathcal{A}_{r-1}$ is $\delta/2$, we obtain that $\Pr\left(\mathcal{A}_{r-1} \text{ errs} \mid \mathcal{E}\right) \leq \delta/2$. By plugging in this bound, together with Eq (6.3.2) and Eq (6.3.3) to Eq (6.3.1), we obtain that $\mathcal{A}_r$ is also a δ-error algorithm, finalizing the proof of induction step. □

Next, we prove an upper bound on the query complexity of $\mathcal{A}_r$ for any $r \geq 1$.

Lemma 6.3.5. *Suppose the input to Algorithm 9 satisfies the assumptions in Lemma 6.3.3; then Algorithm 9 makes at most* $\frac{10m}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log\left(8k/\delta\right)\right)$ *many coin tosses.*

Proof. The proof is again by induction on the number of rounds r. The base case of $r = 1$ is trivially true. Now suppose the bounds are true for all integers smaller than $r \leq \log^*(m) - 3$ and we prove the lemma for the case of r rounds, i.e., for $\mathcal{A}_r$. Note that the total number of coin tosses in $\mathcal{A}_r$ is the sum of coins tosses in step 3 (which is $m \cdot t_r$) and the coins tosses in the recursive call which we bound bellow. For the recursive call there are two cases to consider depending on which of step 12 (Case 1) or step 14 (Case 2) in Algorithm 9 is being executed.

Case 1: In this case $\mathcal{A}_1$ is called with the confidence parameter $\delta/2$ on at most $2k$ coins. We do not use the induction hypothesis here and instead argue directly that,

$$\begin{aligned}
\mathsf{cost}(\mathcal{A}_r) &= m \cdot t_r + \mathsf{cost}(\mathcal{A}_1) \\
&\leq m \cdot t_r + \frac{4k}{\Delta^2} \cdot (\log(2k) + \log(16k/\delta)) \\
&\leq m \cdot t_r + \frac{8k}{\Delta^2} \cdot \log(8k/\delta) \\
&\leq m \cdot t_r + \frac{8m}{\Delta^2} \cdot \log(8k/\delta) && \text{(as } k \leq m\text{)} \\
&= \frac{2m}{\Delta^2} \cdot \left(\mathrm{ilog}^{(r)}(m) + \log(8k/\delta)\right) + \frac{8m}{\Delta^2} \cdot \log(8k/\delta) \\
& && \text{(by plugging in the value of } t_r\text{)} \\
&< \frac{10m}{\Delta^2} \cdot \left(\mathrm{ilog}^{(r)}(m) + \log(8k/\delta)\right)
\end{aligned}$$

which proves the induction step in this case.

Case 2: In this case, $\mathcal{A}_{r-1}$ is called with the confidence parameter $\delta/2$ on at most $\frac{2m}{\mathrm{ilog}^{(r-1)}(m)}$ coins. Hence, by induction, the total number of coin tosses made in recursive calls is

$$\begin{aligned}
\mathsf{cost}(\mathcal{A}_r) &= m \cdot t_r + \mathsf{cost}(\mathcal{A}_{r-1}) \\
&\leq m \cdot t_r + \frac{20m}{\Delta^2 \cdot \mathrm{ilog}^{(r-1)}(m)} \cdot \left(\mathrm{ilog}^{(r-1)}(2m) + \log(16k/\delta)\right) \\
&\leq m \cdot t_r + \frac{20m}{\Delta^2 \cdot \mathrm{ilog}^{(r-1)}(m)} \cdot \left(\mathrm{ilog}^{(r-1)}(m) + 1 + \log(8k/\delta) + 1\right) \\
&< m \cdot t_r + \frac{20m}{\Delta^2} + \frac{22m \cdot \log(8k/\delta)}{\Delta^2 \cdot \mathrm{ilog}^{(r-1)}(m)} \\
&< \frac{2m}{\Delta^2} \cdot \left(\mathrm{ilog}^{(r)}(m) + \log(8k/\delta)\right) + \frac{8m \cdot \mathrm{ilog}^{(r)}(m)}{\Delta^2} + \frac{8m \cdot \log(8k/\delta)}{\Delta^2}
\end{aligned}$$

where in the last inequality we used the bound on t_r plus the fact that $\mathrm{ilog}^{(r)}(m) \geq 16$ as $r \leq \log^*(m) - 3$. This concludes the proof of Lemma 6.3.5. □

Theorem 6.3.1 now follows immediately from Lemma 6.3.3 and Lemma 6.3.5.

6.4 Top-k Ranking from Pairwise Comparisons

The problem of ranking from pairwise comparisons arises in many applications including sports rankings, recommender systems, crowdsourcing and others, and has received increasing attention in recent years (Gleich and Lim, 2011; Jamieson and Nowak, 2011; Negahban et al., 2012; Busa-Fekete et al., 2013; Rajkumar and Agarwal, 2014; Chen and Suh, 2015; Shah and Wainwright, 2015; Jang et al., 2016; Heckel et al., 2016; Braverman et al., 2016a). Here there are n items, and an unknown preference matrix $\mathbf{P} \in [0,1]^{n\times n}$ satisfying $P_{ij} + P_{ji} = 1$ for all $i, j \in [n]$, such that whenever items i and j are compared, item i beats item j with probability P_{ij} and j beats i with probability $P_{ji} = 1 - P_{ij}$. Previous studies have often made strong assumptions on the preference matrix $\mathbf{P}$; here we consider a very general setting where we make no assumptions on $\mathbf{P}$.

We are interested in the problem of identifying the top-k items according to the *Borda score*, which for item i is defined as the probability that i beats another item j drawn uniformly at random:

$$\tau_i = \frac{1}{n-1}\sum_{j\neq i} P_{ij}\,.$$

Ranking according to Borda scores is very natural and encompasses several special cases. For example, Chen and Suh (2015) and Jang et al. (2016) assume $\mathbf{P}$ follows a *Bradley-Terry-Luce* (BTL) model, under which there is a 'score' vector $\mathbf{w} \in \mathbb{R}^n_{++}$ such that $P_{ij} = \frac{w_i}{w_i+w_j}$ $\forall i, j$, and seek to identify the top-k items according to the scores w_i; it can be verified that for such $\mathbf{P}$, ranking by Borda scores is equivalent to ranking by the scores w_i. Feige et al. (1994); Braverman et al. (2016a) assume $\mathbf{P}$ follows a *noisy permutation* model[3], under which there is a permutation $\sigma \in S_n$ and noise parameter $p \in [0, \frac{1}{2})$ such that $P_{ij} = 1 - p$ if $\sigma(i) < \sigma(j)$ and $P_{ij} = p$ otherwise, and seek to identify the top-k items according to σ; again, it can be verified that for such $\mathbf{P}$, ranking by Borda scores is equivalent to ranking according to σ. Here we make no such assumptions on $\mathbf{P}$. The general problem of top-k ranking from pairwise

[3] The results of Feige et al. (1994); Braverman et al. (2016a) can be further extended to a slightly more general model where $\mathbf{P}$ is such that there is a permutation $\sigma \in S_n$ and noise parameter $p \in [0, \frac{1}{2})$ such that $P_{ij} \geq 1 - p$ if $\sigma(i) < \sigma(j)$ and $P_{ij} \leq p$ otherwise.

comparisons under Borda scores has been considered recently by Busa-Fekete et al. (2013), Shah and Wainwright (2015) and Heckel et al. (2016); however, these studies are either in the non-adaptive setting (where pairwise comparisons are observed for randomly drawn item pairs) or in the fully adaptive setting (where one can actively query pairs to be compared with no limit on the number of rounds of adaptivity). Here we consider the limited-adaptivity setting, and show that our results for the coin problem studied in Section 6.2 also yield an optimal algorithm and corresponding lower bound for top-k ranking in this setting.

In order to apply the algorithm of Section 6.2 to the top-k ranking problem, observe that we can view each item i as a coin with bias p_i equal to its Borda score τ_i. In order to toss coin i, we simply select another item $j \in [n] \setminus \{i\}$ uniformly at random, and compare i and j; clearly, this results in a win for item i (heads outcome) with probability τ_i. Thus, the Agressive-Elimination algorithm from Section 6.2 applies directly, with $O(\frac{n}{\Delta_k^2} \log k)$ pairwise comparisons and $\log^*(n)$ rounds of adaptivity. Thus we require fewer comparisons than in the passive setting, and fewer rounds of adaptivity than the previous active algorithms of Busa-Fekete et al. (2013) and Heckel et al. (2016) (see Table 5).

6.5 Extension to Sub-Gaussian Rewards

In this section we discuss the problem of best arms identification in multi-armed bandits with sub-gaussian reward distributions defined as:

Definition 6.5.1. *(Sub-Gaussian Distributions) For any $b > 0$, we say a distribution $\mathcal{D}$ on $\mathbb{R}$ is b-sub-gaussian if for the random variable X drawn from $\mathcal{D}$ and any $t \in \mathbb{R}$, we have that*

$$\mathbb{E}\left[\exp(t \cdot \mathsf{X} - t\,\mathbb{E}[\mathsf{X}])\right] \leq \exp(b^2 \cdot t^2/2)\,.$$

The Bernoulli distribution is a special case of the 1-sub-Gaussian distribution. Any distribution with support in $[0, b]$ is a b-sub-Gaussian distribution. The b-sub-Gaussian family also contains many unbounded distributions such as the Gaussian distribution. We next give a version of Hoeffding's inequality for b-sub-Gaussian distributions.

Lemma 6.5.2. *(Hoeffding's inequality) Let $\mathsf{X}_1, \ldots, \mathsf{X}_\mathsf{m}$ be an i.i.d. sequence of random variables drawn from a b-sub-Gaussian distribution $\mathcal{D}$ with $\mu = \mathbb{E}_{\mathsf{X} \sim \mathcal{D}}[\mathsf{X}]$. Then for any $\epsilon > 0$, we have*

$$\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m}\mathsf{X}_i - \mu\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{m\epsilon^2}{2b^2}\right)$$

We are given n arms, and the reward that we get on pulling each arm is a b-sub-Gaussian random variable with unknown mean. Let μ_i be the mean reward of arm $i \in [n]$. We define the problem of k best arms identification as: given arms $[n]$ with (unknown) mean rewards $\{\mu_i\}_{i=1}^{n}$, a parameter $k \in [n]$, the goal is to identify a set of k best arms in terms of mean rewards. We will assume that the set of k best arms is unique.

For any $0 < \delta < 1$, a δ-error algorithm $\mathcal{A}$ for solving this problem is allowed to *pull* the arms in $[n]$ and based on the outcomes of these pulls, return a set of arms which is the set of top-k arms w.p. at least $1 - \delta$.

We now define the *gap parameter* for an instance of this problem in terms of the differences in mean rewards. For any $i \in [n]$, let,

$$\Delta_i = \begin{cases} \mu_{[i]} - \mu_{[k+1]} & \text{if } i \leq k \\ \mu_{[k]} - \mu_{[i]} & \text{otherwise} \end{cases} .$$

The gap parameter is then Δ_k, which is the difference between the mean rewards of k-th and $(k+1)$-th best arms.

We consider algorithms that in each round chooses a *multi-set* of arms to pull. The choice of this multi-set is *adaptive*, i.e. it is dependent on the history of rewards in previous rounds. Following the coin tossing problem, we denote by $\mathsf{deg}(\mathcal{A})$ the round complexity of algorithm

$\mathcal{A}$, and by $\mathsf{cost}(\mathcal{A})$ the total number of arms pulled. We are interested in algorithms for solving this problem which have low round complexity. In particular, given a parameter r we are interested in δ-error algorithms $\mathcal{A}$ which have $\mathsf{deg}(\mathcal{A}) \leq r$.

We now show that Algorithm 9 can be extended to solve the problem of best-arms identification in multi-armed bandits when the reward distribution is sub-Gaussian. We prove the following theorem:

Theorem 6.5.3. *There exists an algorithm that given any number of rounds $r \geq 1$, integer $k \geq 1$, n arms with b-sub-Gaussian rewards with $b > 0$, and the gap parameter $\Delta_k \in (0, 1)$, and confidence parameter $\delta \in (0, 1)$, finds the set of the k best arms w.p. $1 - \delta$ in r rounds with $O\left(\frac{b^2 n}{\Delta_k^2} \cdot \left(\text{ilog}^{(r)}(n) + \log\left(k/\delta\right)\right)\right)$ pulls.*

To prove the above theorem, the only change required in Algorithm 9 is that the number of pulls in each round also depends on the parameter b of the sub-Gaussian distribution. Specifically, we set

$$t_r := \frac{8b^2}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log\left(8k/\delta\right)\right) ,$$

in step 2 of Algorithm 9, while all the other steps remain the same. We first prove a claim on the estimation of rewards of sub-Gaussian rewards. This is similar to Claim 6.3.4 for the coin problem and we define ϵ in the same way as done in the proof of Theorem 6.3.1.

Claim 6.5.4. *For any round $r \geq 1$, and any arm $i \in S_r$, $\Pr\left(|\widehat{\mu}_i - \mu_i| \geq \varepsilon\right) \leq \frac{\delta}{4k \cdot \text{ilog}^{(r-1)}(m)}$.*

Proof. By Hoeffding's inequality for b sub-Gaussians Lemma 6.5.2, we have,

$$\begin{aligned}\Pr\left(|\widehat{\mu}_i - \mu_i| \geq \varepsilon\right) &\leq 2\exp\left(-\frac{\epsilon^2 \cdot t_r}{2b^2}\right) \\ &\leq 2\exp\left(-\left(\text{ilog}^{(r)}(m) + \log(8k/\delta)\right)\right) \leq \frac{\delta}{4k \cdot \text{ilog}^{(r-1)}(m)}\end{aligned}$$

as $\mathrm{ilog}^{(r)}(m) = \log \mathrm{ilog}^{(r-1)}(m)$. □

The rest of the proof is exactly the same as the proof of Theorem 6.3.1. The lower bound follows from the fact that Bernoulli distributions are a special case of the 1-sub-Gaussian distributions.

6.6 Conclusion

We considered the question of learning with limited rounds of adaptivity in the context of several learning problems: the k most biased coins problem, the closely related k best arms identification problem in stochastic multi-armed bandits (MABs), and top-k ranking from pairwise comparisons. We developed an algorithm which applies to all these problems, and that achieves the optimal worst-case query complexity for these problems in just $\log^*(n)$ rounds of adaptivity, in contrast with previous results which require $\Omega(\log n)$ rounds.

In recent years, there also has been much interest in the MAB literature (and increasingly, in the ranking literature) in adaptive algorithms whose query complexity depends not only on the gap Δ_k between the k-th and $(k+1)$-th best items, but also on the gaps of other items (see Tables 4–5). The optimal query complexity as a function of these parameters, referred to as *instance-wise optimality*, is not yet fully understood despite significant progress in recent years; see, e.g., (Chen and Li, 2015; Chen et al., 2017a) and references therein. The round complexity of the state-of-the-art algorithms (Karnin et al., 2013; Jamieson et al., 2013; Chen and Li, 2015) for this setting has at least a logarithmic dependence on n, as they call the $\log(n)$-round HALVING algorithm of Even-Dar et al. (2006) as a subroutine. It is possible to reduce the round complexity of these algorithms to have a $\log^*$ dependence on n by using an (ϵ, δ)-PAC version[4] of our algorithm as a subroutine instead of HALVING. The round complexity of these algorithms also depends on the gaps Δ_i's, and it is not clear whether the dependence on these Δ_i's is necessary. Closing this gap remains an interesting open question; its resolution would further enhance our understanding of the role of the

[4] Here, the goal is to return a set of k coins whose biases are at least $p_{[k]} - \epsilon$ with probability $\geq 1 - \delta$, for some parameters ϵ, δ. Our algorithm can be easily extended to this (ϵ, δ)-PAC setting.

degree of adaptivity in designing learning algorithms.

Chapter 7

Stochastic Submodular Cover with Limited Adaptivity

In this chapter we continue our discussion at the interface of machine learning and theoretical computer science, and study limited adaptivity for the problem of stochastic submodular cover which has received a lot of interest in both communities.

7.1 Introduction

7.1.1 Background

Submodular functions naturally arise in many applications domains including algorithmic game theory, machine learning, and social choice theory, and have been extensively studied in combinatorial optimization. Many computational problems can be modeled as the *submodular cover* problem where we are given a non-negative monotone submodular function f over a ground set E, and the goal is to choose a smallest subset $S \subseteq E$ such that $f(S) = Q$ where $Q = f(E)$. A well-studied special case is the set cover problem where the function f is the coverage function and the items correspond to subsets of an underlying universe. Even this special case is known to be NP-hard to approximate to a factor better than $\Omega(\log Q)$ (Dinur and Steurer, 2014; Feige, 1998; Lund and Yannakakis, 1994; Moshkovitz, 2015), and on the other hand, the classic paper of Wolsey (Wolsey, 1982) shows that the problem admits a poly-time $O(\log Q)$-approximation for any integer-valued monotone submodular function.

In this chapter we consider the *stochastic version* of the problem that naturally arises when there is uncertainty about items. For instance, in stochastic influence spread in networks, the set of nodes that can be influenced by any particular node is a random variable whose value depends on the realized state of the influencing node (e.g. being successfully activated). In sensor placement problems, each sensor can fail partially or entirely with certain probability and the coverage of a sensor depends on whether the sensor failed or not. In data acquisition for machine learning (ML) tasks, each data point is apriori a random variable that can take different values, and one may wish to build a dataset representing a diverse set of values. For example, if one wants to build a ML model for identifying a new disease from gene patterns, one would start by building a database of gene patterns associated to that disease. In this case, each person's gene pattern is a random variable that can realize to different values depending on the race, gender, etc. For other examples, we refer the reader to Liu et al. (2008) (application in databases) and Anagnostopoulos et al. (2015) (application in document retrieval).

In the *stochastic submodular cover* problem, we are given m stochastic items which are different random variables that independently realize to an element of E, and the goal is to find a lowest cost set of stochastic items whose realization R satisfies $f(R) = Q$. In network influence spread problems each item corresponds to a node in the network, and its realization corresponds to the set of nodes it can influence. In sensor placement problems an item corresponds to a sensor and its realization corresponds to the area that it covers upon being deployed. In the case of data acquisition, an item corresponds to a data point and its realization corresponds to the value it takes upon being queried. The problem captures as a special case the stochastic set cover problem and more generally, stochastic covering integer programs.

In stochastic optimization, a powerful computational resource is *adaptivity*. An *adaptive* algorithm for stochastic submodular cover chooses an item to realize and based on its realization, decides which item to realize next. A *non-adaptive* algorithm on the other hand needs to choose a permutation of items and realize them in the order specified by the permutation until the function value reaches Q. The cost of the algorithm in both cases is the number (or costs) of items realized by the algorithm. It is well-understood that in general, adaptive algorithms perform better than non-adaptive algorithms in terms of cost of coverage. However, in practical applications a non-adaptive algorithm is better from the point of view of practitioners as it eliminates the need of sequential decision making and instead requires them to make just one decision. This motivates the study of separation between the performance of adaptive and non-adaptive algorithms, known as the *adaptivity gap*. For many stochastic packing problems, the adaptivity gap is only a constant. For instance, the adaptivity gap for budgeted stochastic max coverage where you are given a constraint on the number of items that can be chosen and the goal is to maximize coverage, the adaptivity gap is bounded by $1 - 1/e$ (Asadpour et al., 2008). In a sharp contrast, for the covering version of the problem, it is not difficult to show an adaptivity gap of $\Omega(Q)$ (Goemans and Vondrák, 2006).

Motivated by this striking separation between the power of adaptive and non-adaptive algorithms, we consider the following question in this chapter: does one need full power of adaptivity to obtain a near-optimal solution to stochastic submodular cover? In particular, how does the performance guarantees change when an algorithm interpolates between these two extremes using a few rounds of adaptivity.

7.1.2 Our Contributions

We define an *r-round adaptive* algorithm to be an algorithm that chooses a permutation of all available items in each round $k \in [r]$, and a threshold τ_k, and realizes items in the order specified by the permutation until the function value is at least τ_k. A non-adaptive algorithm would then correspond to the case $r = 1$ (with $\tau_1 = Q$), and an adaptive algorithm would correspond to the case $r = m$ (with $\tau_k = 0$ for all $k \in [r]$). The permutation for each round k is chosen adaptively based on the realization in the previous rounds, but the ordering inside each round remains fixed regardless of the realizations seen inside the round. We will call this the "permutation framework" for an r-round algorithm.

Our main result is that for any integer r, there exists a poly-time r-round adaptive algorithm for stochastic submodular cover whose expected cost is $\widetilde{O}(Q^{1/r})$ times the expected cost of a fully adaptive algorithm, where the $\widetilde{O}$ notation is hiding a logarithmic dependence on the number of items and the maximum cost of any item. Prior to our work, such a result was not known even for the case of $r = 1$ and when f is the coverage function. Indeed achieving such a result was cast as an open problem by Goemans and Vondrak (Goemans and Vondrák, 2006) who achieved an $O(n^2)$ bound (corresponding to $O(Q^2)$) on the adaptivity gap of stochastic set cover. Furthermore, we show that for any r, there exist instances of the stochastic submodular cover problem where no r-round adaptive algorithm can achieve better than $\Omega(Q^{1/r})$ approximation to the expected cost of a fully adaptive algorithm. Our lower bound result holds even for coverage function and for algorithms with unbounded computational power. Thus our work shows that logarithmic rounds of adaptivity are necessary and sufficient to obtain near-optimal solutions to the stochastic submodular cover problem, and even few rounds of adaptivity are sufficient to sharply reduce the adaptivity gap.

Remark 7.1.1. One may consider an alternate notion of r-round adaptive algorithm: In each round $k \in [r]$, the algorithm chooses a fixed set of items to realize in parallel where the choice of the set depends on the realizations in the previous rounds (instead of a permutation over items). Let us call this framework the "set framework". One benefit of this variation is that items in each round can be realized in parallel. Unfortunately in this framework, any algorithm that always outputs a valid cover (as is our requirement), must in general include all remaining items in the last round, because for any proper subset of the remaining items there will be positive probability that this subset will not able to cover the entire set. Hence, the r-round adaptivity gap would be $\Omega(m)$.

Hence, one would have to consider a relaxed version of the problem and require that the algorithm achieves the desired coverage guarantee only with probability $1 - o(1)$. Our algorithmic results directly carry over to this variant of the problem. In particular, for any fixed r, we obtain poly-time r-round adaptive algorithm in the set framework whose cost is $\widetilde{O}(Q^{1/r})$ times the expected cost of a fully adaptive algorithm, and that succeeds with probability at least $1 - o(1)$. At the same time, our lower bound of $\Omega(Q^{1/r})$ continues to hold in this relaxed setting. In the following we will provide results for only the permutation framework, with the understanding that all our results carry over to the set framework with the relaxed version of the problem.

7.1.3 Related Work

The problem of submodular cover was perhaps first studied by Wolsey (1982), who showed that a *greedy* algorithm achieves an approximation ratio of $\log(Q)$. Subsequent to this there has been a lot of work on this problem in various settings (Golovin and Krause, 2010; Azar and Gamzu, 2011; Azar et al., 2009; Im et al., 2016; Deshpande et al., 2014; Grammel et al., 2016; Kambadur et al., 2017). To our knowledge, the question of adaptivity in stochastic covering problems was first studied in Goemans and Vondrák (2006) for the special case of stochastic set cover and covering integer programs. It was shown that the adaptivity gap of this problem is $\Omega(n)$, where n is the size of the universe to be covered. A non-adaptive algorithm for this problem with an adaptivity gap of $O(n^2)$ was also presented.

Subsequently there has been a lot of work on stochastic set cover and the more general stochastic submodular cover problem in the fully adaptive setting. A special case of stochastic set cover was studied by Liu et al. (2008) in the adaptive setting, and an *adaptive greedy algorithm* was studied[1]. In Golovin and Krause (2010) the notion of "adaptive submodularity" was defined for adaptive optimization, which demands that given any partial realization of items, the marginal function with respect to this realization remains monotone submodular. This paper also presented an *adaptive greedy algorithm* for the problem of stochastic submodular cover, and stochastic submodular maximization subject to cardinality constraints.[2] In Im et al. (2016) a more general version of stochastic submodular cover problem was studied in the fully adaptive setting, and their results imply the best-possible approximation ratio of $\log(Q)$ for stochastic submodular cover. In Deshpande et al. (2014) an *adaptive dual greedy* algorithm was presented for this problem. It was also shown that the *adaptive greedy algorithm* of Golovin and Krause (2010) achieves an approximation ratio of $k\log(P)$, where

[1] The paper originally claimed an approximation ratio of $\log(n)$ for this algorithm, however, the claim was later retracted by the authors due to an error in the original analysis (Parthasarathy, 2018)

[2] It was originally claimed that this algorithm achieves an approximation ratio of $\log(Q)$ where Q is the desired coverage, however, the claim was later retracted due to an error in the analysis (Nan and Saligrama, 2017). The authors have claimed an approximation ratio of $\log^2(Q)$ since then.

P is the maximum function value any item can contribute, and k is the maximum support size of the distribution of any item. There has also been work on this problem when the realization of items can be correlated, unlike our setting where the realization of each item is independent. In this setting, Kambadur et al. (2017) gives an adaptive algorithm which achieves an approximation ratio of $\log(Qs)$, where Q is the desired coverage, and s denote the support size of the joint distribution of these correlated items. In the case of independent realizations this quantity will typically be exponential in the number of items. In Grammel et al. (2016) a similar result was shown for a slightly different algorithm.

The question of adaptivity has also been studied for a related problem of *stochastic submodular maximization* subject to cardinality constraints (Asadpour et al., 2008). The goal in this problem is to find a set of items with cardinality at most k, so as to maximize the expected value of a stochastic submodular function. This paper showed that a non-adaptive greedy algorithm for this problem achieves an approximation ratio of $(1 - \frac{1}{e})^2$ with respect to an optimal adaptive algorithm. This result was later generalized to stochastic submodular maximization subject to matroid constraints (Asadpour and Nazerzadeh, 2016). In Gupta et al. (2017), the adaptivity gap of stochastic submodular maximization subject to a variety of *prefix-closed constraints* was studied under the setting where the distribution of each item is Bernoulli. This class of prefix-closed constraints includes matroid and knapsack constraints among others. It was shown that there is a non-adaptive algorithm that achieves an approximation ratio of $1/3$ with respect to an optimal adaptive algorithm. In Hellerstein et al. (2015), the problem of stochastic submodular maximization was also studied under various types of constraints, including knapsack constraints. An approximation ratio of τ for this problem under knapsack constraint was given, where τ is the smallest probability of any element in the ground set being realized by any item. The question of adaptivity has also been studied for other stochastic problems such as stochastic packing, knapsack, matching etc. (see, e.g. Dean et al. (2005, 2008); Yamaguchi and Maehara (2018); Blum et al. (2015); Assadi et al. (2017, 2016) and references therein).

There has also been a lot of work under the framework of 2-stage or multi-stage stochastic programming (Shapiro et al., 2009; Swamy and Shmoys, 2012; Charikar et al., 2005; Shmoys and Swamy, 2004). In this framework, one has to make sequential decisions in a stochastic environment, and there is a parameter λ, such that the cost of making the same decision increases by a factor λ after each stage. The stochastic program in each stage is defined in terms of the expected cost in the later stages. The central question in these problems is– when can we find good solutions to this complex stochastic program, either by directly solving it or by finding approximations to it? This largely depends on the complexity of the stochastic program at hand. For example, if the distribution of the environment is explicitly given, then one might be able to solve the stochastic program exactly by using integer programming, and this question becomes largely *computational* in nature. This is fundamentally different than the *information theoretic* question we consider in this chapter.

Aside from the stochastic setting, algorithms with limited adaptivity have been studied across a wide spectrum of areas in computer science including in sorting and selection (e.g. Valiant (1975); Cole (1986); Braverman et al. (2016b)), multi-armed bandits (e.g. Perchet et al. (2015a); Agarwal et al. (2017a)), algorithms design (e.g. Balkanski and Singer; Emamjomeh-Zadeh et al. (2016); Ene and Nguyen (2018); Balkanski et al. (2018); Balkanski and Singer (2020); Breuer et al. (2020); Fahrbach et al. (2019)), among others; we refer the interested reader to these papers and references therein for more details.

Remark 7.1.2. Our study of r-round adaptive algorithm for submodular cover is reminiscent of a recent work of Chakrabarti and Wirth (2016) on multi-pass streaming algorithms for the set cover problem. They showed that allowing additional passes over the input in the streaming setting (similar-in-spirit to more rounds of adaptivity) can significantly improve the performance of the algorithms and established tight pass-approximation tradeoffs that are similar (but not identical) to r-round adaptivity gap bounds in Results 1 and Results 2. In terms of techniques, our upper bound result—our main contribution—is almost entirely disjoint from the techniques in Chakrabarti and Wirth (2016) (and works for the more general problem of submodular cover, whereas the results in Chakrabarti and Wirth (2016) are specific to set cover), while our lower bound uses similar instances as Chakrabarti and Wirth (2016) but is based on an entirely different analysis.

7.1.4 Organization

In Section 7.2 we introduce the problem more formally. In Section 7.3 we provide an overview of our technical results. In Section 7.4 we present some preliminaries for our problem. In Section 7.5 we present a technical overview of our main results. In Section 7.6 we present a non-adaptive selection algorithm that will be used to prove our upper bound result in Section 7.7. We present the lower bound result in Section 7.8.

7.2 Problem Statement

Let $\mathsf{X} := \{\mathsf{X}_1, \ldots, \mathsf{X}_m\}$ be a collection of m *independent* random variables each supported on the same ground set E and f be an integer-valued[3] non-negative monotone submodular function $f : 2^E \rightarrow \mathbb{N}_+$. We will refer to random variables X_i's as items and any set $\mathcal{S} \subseteq \mathsf{X}$ as a set of items. For any $i \in [m]$, we use $x_i \in E$ to refer to a realization of item (random variable) X_i and define $X := \{x_1, \ldots, x_m\}$ as the realization of X. We slightly abuse notation[4] and extend f to the ground set of items X such that for any set $\mathcal{S} \subseteq \mathsf{X}$, $f(\mathcal{S}) := f(\cup_{\mathsf{X}_i \in \mathcal{S}} \mathsf{X}_i)$: this definition means that for any realization S of $\mathcal{S}$, $f(S) = f(\cup_{x_i \in S} x_i)$. Finally, there is an integer-valued cost $c_i \in [C]$ associated with item $\mathsf{X}_i \in \mathsf{X}$.

Let $Q := f(E)$. For any set of items $\mathsf{S} \subseteq \mathsf{X}$, we say that a realization S of S is *feasible* iff $f(S) = Q$. We will assume that any realization X of $\mathcal{X}$ is always feasible, i.e. $f(X) = Q$[5]. We will say that a realization X of $\mathcal{X}$ is *covered* by a realization $S \subseteq X$ of $\mathcal{S}$ iff S is feasible. The goal in the stochastic submodular cover problem is to find a set of items $\mathsf{S} \subseteq \mathsf{X}$ with the minimum cost which gets realized to a feasible set. In order to do so, if we include any item

[3] We present our results for integer-valued functions for simplicity of exposition. All our results can easily be generalized to positive real-valued functions.

[4] Note that here $f : 2^E \rightarrow \mathbb{N}_+$ is being extended to a function $f' : 2^{\mathsf{X}} \rightarrow \mathbb{N}_+$, but we chose to refer to f' as f.

[5] One can ensure this by adding an item $\mathcal{X}_i$ to the ground set such that $f(x_i) = Q$ for all realizations x_i of $\mathcal{X}_i$, but cost of this item is higher than the combined cost of all other items.

X_i to S we pay a cost c_i, and once included, X_i would be realized to some $x_i \in E$ and is fixed from now on. Once a decision made regarding inclusion of an item in S, this item cannot be removed from S.

For any set of items $\mathcal{S} \subseteq \mathsf{X}$, we define $\mathsf{cost}(\mathcal{S})$ to be the total cost of all items in $\mathcal{S}$, i.e. $\mathsf{cost}(\mathcal{S}) = \sum_{i \in [m]} c_i \cdot \mathbf{1}[\mathsf{X}_i \in \mathcal{S}]$, where $\mathbf{1}[\cdot]$ is an indicator function. For any algorithm $\mathcal{A}$, we refer to the total cost of solution $\mathcal{S}$ returned by $\mathcal{A}$ on an instantiation X of X as the *cost* of $\mathcal{A}$ on X denoted by $\mathsf{cost}(\mathcal{A}(X))$. We are interested in *minimizing* the *expected* cost of the algorithm $\mathcal{A}$, i.e., $\mathbb{E}_{X \sim \mathsf{X}}[\mathsf{cost}(\mathcal{A}(X))]$.

Example 7.2.1 (**Stochastic Set Cover**). *A canonical example of the stochastic submodular cover problem is the stochastic set cover problem. Let U be a universe of n "elements" (not to be mistaken with "items") and $\mathsf{X} = \{\mathsf{X}_1, \ldots, \mathsf{X}_m\}$ be a collection of m random variables where each random variable X_i is supported on subsets of U, i.e., realizes to some subset $T_i \subseteq U$. We refer to each random variable X_i as a* stochastic set*. In the stochastic set cover problem, the goal is to pick a smallest (or minimum weight) collection S of items (or equivalently sets) in X such that the realized sets in this collections cover the universe U.*

We consider the following types of algorithms (sometimes referred to as policies in the literature) for the stochastic submodular cover problem:

- **Non-adaptive:** A non-adaptive algorithm simply picks a fixed ordering of items in X and insert the items one by one to S until the realization S of S become feasible.

- **Adaptive:** An adaptive algorithm on the other hand picks the next item to be included in S adaptively based on the realization of previously chosen items. In other words, the choice of each item to be included in S is now a function of the realization of items already in S.

- ***r*-round adaptive:** We define r-round adaptive algorithms as an "interpolation" between the above two extremes. For any integer $r \geq 1$, an r-round adaptive algorithm chooses the items to be included in S in r rounds of adaptivity: In each round $i \in [r]$, the algorithm chooses a *threshold* $\tau_i \in \mathbb{N}_+$ and an ordering over items, and then inserts the items one by one according to this ordering to S until for the realized set S, $f(S) \geq \tau_i$. Once this round finishes, the algorithm decides on an ordering over the remaining items *adaptively* based on the current realization.

In above definitions, a non-adaptive algorithm corresponds to case of $r = 1$ round adaptive algorithm (with $\tau_1 = Q$) and a (fully) adaptive algorithm corresponds to the case of $r = m$ (here τ_i is irrelevant and can be thought as being zero).

Adaptivity gap. We use OPT to refer to the optimal adaptive algorithm for the stochastic submodular cover problem, i.e., an adaptive algorithm with minimum expected cost. We use the expected cost of OPT as the main benchmark against which we compare the cost of other algorithms. In particular, we define *adaptivity gap* as the ratio between the expected cost of the best non-adaptive algorithm for the submodular cover problem and the expected cost of OPT. Similarly, for any integer r, we define the *r-round adaptivity gap* for r-rounds adaptive algorithms in analogy with above definition.

Remark 7.2.2. The notion of "best" non-adaptive or r-round adaptive algorithm defined above allow *unbounded computational* power to the algorithm. Hence, the only limiting factor of the algorithm is the *information-theoretic* barrier caused by the *uncertainty* about the underlying realization.

7.3 Overview of Results

In this chapter, we establish *tight* bounds (up to logarithmic factor) on the r-round adaptivity gap of the stochastic submodular cover problem for any integer $r \geq 1$. Our main result is an r-round adaptive algorithm (for any integer $r \geq 1$) for the stochastic submodular cover problem.

Result 1 (**Main Result**)**.** *For any integer $r \geq 1$ and any monotone submodular function f, there exists an r-round adaptive algorithm for the stochastic submodular cover problem for function f and set of items $\mathcal{X}$ with cost of each item bounded by C that incurs expected cost $O(Q^{1/r} \cdot \log Q \cdot \log(mC))$ times the expected cost of the optimal adaptive algorithm.*

A corollary of Result 1 is that the r-round adaptivity gap of the submodular cover problem is $\widetilde{O}(Q^{1/r})$. This implies that using only $O\left(\frac{\log Q}{\log\log Q}\right)$ rounds of adaptivity, one can reduce the cost of the algorithm to within *poly-logarithmic* factor of the optimal adaptive algorithm. In other words, one can "harness" the (essentially) full power of adaptivity, in only logarithmic number of rounds.

Various stochastic covering problems can be cast as submodular cover problem, including the stochastic set cover problem and the stochastic covering integer programs studied previously in the literature (Goemans and Vondrák, 2006; Golovin and Krause, 2010; Deshpande et al., 2014). As such, Result 1 directly extends to these problems as well. In particular, as a (very) special case of Result 1, we obtain that the adaptivity gap of the stochastic set cover problem is $\widetilde{O}(n)$ (here n is the size of the universe), improving upon the $O(n^2)$ bound of Goemans and Vondrak (Goemans and Vondrák, 2006) and settling an open question in their work regarding the adaptivity gap of this problem (an $\Omega(n)$ lower bound was already shown in Goemans and Vondrák (2006)).

We further prove that the r-round adaptivity gaps in Result 1 are almost *tight* for any $r \geq 1$.

Result 2. *For any integer $r \geq 1$, there exists a monotone submodular function $f : 2^E \to \mathbb{N}_+$, in particular a coverage function, with $Q := f(E)$ such that the expected cost of any r-round adaptive algorithm for the submodular cover problem for function f, i.e., the stochastic set cover problem, is $\Omega(\frac{1}{r^3} \cdot Q^{1/r})$ times the expected cost of the optimal adaptive algorithm.*

Result 2 implies that the r-round adaptivity gap of the submodular cover problem is $\Omega(\frac{1}{r^3} \cdot Q^{1/r})$, i.e., within poly-logarithmic factor of the upper bound in Result 1. An immediate corollary of this result is that $\Omega(\frac{\log Q}{\log\log Q})$ rounds of adaptivity are necessary for reducing the cost of the algorithms to within logarithmic factors of the optimal adaptive algorithm. We further point out that interestingly, the optimal adaptive algorithm in instances created in Result 2 only requires $r+1$ rounds; as such, Result 2 in fact is proving a lower bound on the gap between the cost of r-round and $(r+1)$-round adaptive algorithms.

We remark that our algorithm in Result 1 is *polynomial time* (for polynomially-bounded item costs), while the lower bound in Result 2 holds again algorithms with unbounded computational power (see Remark 7.2.2).

7.4 Preliminaries

Notation. Throughout this chapter we will use symbols S, T, and R to denote subsets of the ground set E, and use symbols A and B to denote subsets of $[m]$, i.e., indices of items. We will also use symbols $\mathcal{S}, \mathsf{T}$ and $\mathbb{R}$ to denote subsets of $\mathcal{X}$ which realize to subsets S, T and R of the ground set E.

Submodular Functions: Let E be a finite ground set and $\mathbb{N}_+$ be the set of non-negative integers. For any set function $f : 2^E \to \mathbb{N}_+$, and any set $S \subseteq E$, we define the *marginal contribution* to f as the set function $f_S : 2^E \to \mathbb{N}_+$ such that for all $T \subseteq E$,

$$f_S(T) = f(S \cup T) - f(S).$$

When clear from the context, we abuse the notation and for $e \in E$, we use $f(e)$ and $f_S(e)$ instead of $f(\{e\})$ and $f_S(\{e\})$.

A set function $f : 2^E \to \mathbb{N}_+$ is submodular iff for all $S \subseteq T \subseteq E$ and $e \in E$: $f_S(e) \geq f_T(e)$. Function f is additionally monotone iff $f(S) \leq f(T)$. Throughout the chapter, we solely focus on monotone submodular functions unless stated explicitly otherwise.

We use the following two well-known facts about submodular functions throughout the chapter.

Fact 7.4.1. *Let $f(\cdot)$ be a monotone submodular function, then:*

$$\forall S, T \subseteq E \qquad f(S) \leq f(T) + \sum_{e \in S \setminus T} f_T(e).$$

Fact 7.4.2. *Let $f(\cdot)$ be a monotone submodular function, then for any $S \subseteq E$, $f_S(\cdot)$ is also monotone submodular.*

7.5 Technical Overview

We give here an overview of the techniques used in our upper and lower bound results.

7.5.1 Upper Bound on r-round Adaptivity Gap

In this discussion we focus mainly on our non-adaptive ($r = 1$) algorithm, which already deviates significantly from the previous work of Goemans and Vondrak (Goemans and Vondrák, 2006). A non-adaptive algorithm simply picks a permutation of items and realize them one by one in a set S until $f(S) = Q$. Hence, the "only" task in designing a non-adaptive algorithm is to find a "good" ordering of items, that is, an ordering such that its prefix that covers Q has a low expected cost.

Consider the following algorithmic task: In the setting of stochastic submodular cover problem, suppose we are given a (ordered) set S of stochastic items. Can we pick a low-cost (ordered) set T of stochastic items non-adaptively (without looking at a realization of S or T) so that the coverage of $\mathsf{S} \cup \mathsf{T}$ is sufficiently larger than S, i.e., $\mathbb{E}\left[f_{\mathsf{S}}(\mathsf{T})\right]$ is large? Assuming we can do this, we can use this primitive to find sets with large coverage non-adaptively and iteratively, by starting from the empty-set and using this primitive to increase the coverage further repeatedly.

Recall that in the non-stochastic setting, the greedy algorithm is precisely solving this problem, i.e., finds a set T such that $\frac{f_S(T)}{\mathsf{cost}(T)} \geq \frac{Q - f(S)}{\mathsf{cost}(\text{OPT})}$, where with a slight abuse of notation, $\textsc{opt}$ here denotes the optimal non-stochastic cover of $f(E)$. This suggests that one can always find a "low" cost set T with a large marginal contribution to S. For the stochastic problem, however, it is not at all clear whether there always exists a "low cost" (compared to adaptive $\textsc{opt}$) T whose expected marginal contribution to $\mathcal{S}$ is large. This is because there are many different realizations possible for S, and each realization S, in principle may require a *dedicated* set of items $\mathsf{T}(S)$ to achieve a large value $\mathbb{E}\left[f_S(\mathsf{T}(S)) \mid S\right]$. As such, while adaptive $\textsc{opt}$ can first discover the realization S of S and based on that choose $\mathsf{T}(S)$ to increase the expected coverage, a non-adaptive algorithm needs to instead pick $\cup_{S \in \mathsf{S}} \mathsf{T}(S)$, which can have a much larger cost (but the same marginal contribution). This suggests that cost of non-adaptive algorithm can potentially grow with the size of all possible realizations of S. We point out that this task remains challenging even if all remaining inputs other than S are non-stochastic, i.e., always realize to a particular item.

Nevertheless, it turns out that no matter the size of the set of all realizations of S, one can always find a set of stochastic items T such that $\mathbb{E}\left[f_{\mathsf{S}}(\mathsf{T})\right] = \Omega(1) \cdot \mathbb{E}\left[Q - f(\mathsf{S})\right]$ while $\mathsf{cost}(\mathsf{T}) = \widetilde{O}(Q) \cdot \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$, i.e., achieve a marginal contribution proportional to $\mathbb{E}\left[Q - f(\mathsf{S})\right]$ while paying cost which is $\widetilde{O}(Q)$ times larger than $\textsc{opt}$ (here $\textsc{opt}$ corresponds to an optimal adaptive algorithm corresponding the residual problem of covering $Q - f(\mathsf{S})$). Compared to the non-stochastic setting, this cost is $\widetilde{O}(Q)$ times larger than the analogous cost in the non-stochastic setting (see Example 7.6.1). This part is one of the main technical ingredients of our chapter (see Theorem 7.6.2). We briefly describe the main ideas behind this proof.

The idea behind our algorithm is to *sample* several realizations $S_1, \ldots, S_\Psi$ from S and pick a low cost dedicated set T_i for each S_i such that $\mathbb{E}\left[f_{S_i}(\mathsf{T}_i)\right]$ is large (here, the randomness is only on realizations of T_i). This step is quite similar to solving the non-adaptive submodular maximization problem with knapsack constraint for which we design a new algorithm based on an adaptation of Wolsey's LP (Wolsey, 1982) (see Theorem 7.6.3 and discussion before that for more details and comparison with existing results). This allows us to bound the cost of each set T_i by $O(\mathbb{E}[\mathsf{cost}(\textsc{opt})])$. The final (ordered) set returned by this algorithm is then $\mathsf{T} := \mathsf{T}_1 \cup \ldots \cup \mathsf{T}_\Psi$. The ordering within items of T does not matter.

The *main* step of this argument is however to bound the value of Ψ, i.e., the number of samples, by $O(Q)$. This step is done by bounding the total contribution of sets $\mathsf{T}_1, \ldots, \mathsf{T}_\Psi$ on their own, i.e., $\mathbb{E}[f(\mathsf{T}_1 \cup \ldots \cup \mathsf{T}_\Psi)]$ independent of the set S. The intuition is that if we choose, say T_1, with respect to some realization S of S, but T_1 does not have a marginal contribution to most realizations S' of S, then this means that by picking another set T_2, the set $\mathsf{T}_1 \cup \mathsf{T}_2$ needs to have a coverage larger than both T_1 and T_2. As a result, if we repeat this process sufficiently many times, we should eventually be able to increase $\mathbb{E}[f_{\mathsf{S}}(\mathsf{T})]$, simply because otherwise $f(\mathsf{T}) > Q$, a contradiction.

We now use this primitive to design our non-adaptive algorithm as follows: we keep adding set of items to the ordering using the primitive above in *iterative phases.* In each phase p, we run the above primitive multiple times to find a set S_p with $\mathbb{E}[Q - f(\mathsf{S}_p) \mid \mathcal{E}_{p-1}] = o(1)$, where $\mathcal{E}_{p-1}$ is the event that the realization of items picked in previous phases of the algorithm did not cover Q entirely. We further bound the cost of the set S_p with the expected cost of $\textsc{opt}$ conditioned on the event $\mathcal{E}_{p-1}$, i.e., $\mathbb{E}[\mathsf{cost}(\textsc{opt}) \mid \mathcal{E}_{p-1}]$. Notice that this quantity can potentially be much larger than the expected cost of $\textsc{opt}$. However, since the probability that in the permutation returned by the non-adaptive algorithm, we ever need to realize the sets in S_p is bounded by $\Pr(\mathcal{E}_{p-1})$, we can *pay* for the cost of these sets in expectation. By repeating these phases, we can reduce the probability of not covering Q *exponentially fast* and finalize the proof.

We then extend this algorithm to an r-round adaptive algorithm for any $r \geq 1$. For simplicity, let us only mention the extension to 2 rounds (extending to r is then straightforward). We spend the first round to find a (ordered) set S with $f(S) \geq Q - \sqrt{Q}$ with high probability for any realizations S of S. We extend our main primitive above to ensure that if $\mathbb{E}[Q - f(\mathsf{S})] \geq \sqrt{Q}$, then we can find a set T with $\mathbb{E}[f_{\mathsf{S}}(\mathsf{T})] = \Omega(1) \cdot \mathbb{E}[Q - f(\mathsf{S})]$ and $\mathsf{cost}(\mathsf{T}) = \widetilde{O}(\sqrt{Q}) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})]$ (as opposed to $O(Q)$ in the original statement). This is achieved by the fact that when the deficit $Q - f(\mathsf{S})$ is sufficiently large then the rate of coverage per cost is higher, as opposed to when the deficit $Q - f(\mathsf{S})$ is very small. Precisely, we exploit the fact that the gap of $Q - f(\mathsf{S})$ is sufficiently large to reach the contradiction in the original argument with only $O(\sqrt{Q})$ sets $\mathsf{T}_1, \mathsf{T}_2, \ldots$. We then run the previous algorithm using this primitive by setting the threshold $\tau_1 = Q - \sqrt{Q}$. In the next round, we simply run our previous algorithm on the function $f_S(\cdot)$ where S is the realization in the first round. As $f_S(\cdot)$ has maximum value at most $O(\sqrt{Q})$, by the previous argument we only need to pay $\widetilde{O}(\sqrt{Q})$ times expected cost of OPT, hence our total cost is $\widetilde{O}(\sqrt{Q}) \cdot \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$. Extending this approach to r-round algorithms is now straightforward using similar ideas as the thresholding greedy algorithm for set cover (see, e.g. Cormode et al. (2010)).

7.5.2 Lower Bound on Adaptivity Gap

We prove our lower bound for the stochastic set cover problem, a special case of stochastic submodular cover problem (see Example 7.2.1). Let us first sketch our lower bound for two round algorithms. Let $:= \{U_1, \ldots, U_k\}$ be a collection of $k = \text{poly}(n)$ sets to be determined later (recall that n is the size of the universe U we aim to cover). Consider the following instance of stochastic set cover: there exists a single stochastic set T which realizes to one set chosen uniformly at random from sets $\overline{U_1}, \ldots, \overline{U_k}$, i.e., complements of the sets in . We further have k additional stochastic sets where T_i realizes to $U_i \setminus \{e\}$ for e chosen uniformly at random from U_i. Finally, for any element $e \in U$, we have a set T_e with only one realization which is the singleton set $\{e\}$ (i.e., T_e always covers e).

Consider first the following adaptive strategy: pick T in the first round and see its realization, say, $\overline{U_i}$. Pick T_i in the second round and see its realization, say $U_i \setminus \{e\}$. Pick T_e in the third round. This collection of sets is $(U \setminus U_i) \cup (U_i \setminus \{e\}) \cup (\{e\}) = U$, hence it is a feasible cover. As such, in only 3 rounds of adaptivity, we were able to find a solution with cost only 3.

A two-round algorithm is however one round short of following the above strategy. One approach to remedy this would be try to make a "shortcut" by picking more than one sets in each round of this process, e.g., pick the set T_i also in the first round. However, it is easy to see that as long as we do not pick $\Omega(k)$ sets in the first round, or $\Omega(|U_i|)$ sets in the second round, we have a small chance of making such a shortcut. We are not done yet as it is possible that the algorithm covers the universe using entirely different sets (i.e., do not follow this strategy). To ensure that cannot help either, we need the sets in $U_1, \ldots, U_k$ to have "minimal" intersection; this in turns limits the size of each set U_i and hence the eventual lower bound we obtain using this argument.

We design a family of instances that allows us to extend the above argument to r-round adaptive algorithms. We construct these instances using the *edifice* set-system of Chakrabarti and Wirth (2016) that poses a "near laminar" property, i.e., any two sets are either subset-superset of one another or have "minimal" intersection. We remark that this set-system was originally introduced by Chakrabarti and Wirth (2016) for designing multi-pass streaming lower bounds for the set cover problem. While the instances we create in this work are similar to the instances of Chakrabarti and Wirth (2016), the proof of our lower bound is entirely different (lower bound of Chakrabarti and Wirth (2016) is proven using a reduction in communication complexity).

7.6 The Non-Adaptive Selection Algorithm

We introduce a key primitive of our approach in this section for solving the following task: Suppose we have already chosen a subset $\mathsf{S} \subseteq \mathsf{X}$ of items but we are not aware of the realization of these items; our goal is to non-adaptively add another set T to S to increase its expected coverage. Formally, given any monotone submodular function $g : 2^E \rightarrow \mathbb{N}_+$, let $Q_g := g(E)$ be the required coverage on g. Also, for any realization S of S, we use $\Delta(S) := Q_g - g(S)$ to refer to the *deficit* in covering Q_g, and denote by $\Delta := \mathbb{E}\left[\Delta(\mathsf{S})\right]$ the expected deficit of set S. Our goal is now to add (still non-adaptively) a "low-cost" (compared to adaptive $\textsc{opt}$) set T to S to decrease the *expected* deficit. It is easy to see that such a primitive would be helpful for finding sets with "large" coverage non-adaptively and iteratively, by starting from the empty-set and use this primitive to reduce the deficit further by picking another set and then repeat the process starting from this set.

Let us start by giving an example which shows some of the difficulty of this task.

Example 7.6.1. *Consider an instance of stochastic set cover: there exists a single set, say* X_1 *which realizes to* $U \setminus \{e^*\}$ *for an element* e^* *chosen uniformly at random from* U *and* n *singleton sets* $\mathsf{X}_2, \cdots \mathsf{X}_{n+1}$*, each covering a unique element in* U*. If we have already chosen* X_1*, and want to chose more sets in order to decrease the expected deficit, then it is easy to see that even though the cost of* OPT *is only* 2*, no collection of* $o(n)$ *sets can decrease the expected deficit by one. This should be contrasted with the non-stochastic setting in which there always exists a* single *set that reduces a deficit of* Δ *by* $\Delta/\mathsf{cost}(\textsc{opt})$*.*

We are now ready to state our main result in this section.

Theorem 7.6.2. *Let* X *be a collection of items, and let* g *be any monotone submodular function such that* $g(X) = Q_g$ *for every realization* X *of* $\mathcal{X}$*. Let* $\mathsf{S} \subseteq \mathsf{X}$ *be any subset of items and define* $\Delta := \mathbb{E}\left[Q_g - g(\mathsf{S})\right]$*. Given any parameter* $\alpha \geq Q_g/\Delta$*, there is a* randomized non-adaptive *algorithm that outputs a set* $\mathsf{T} \subseteq \mathsf{X} \setminus \mathsf{S}$ *such that cost of* T *is* $O(\alpha) \cdot \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$ *in expectation over the randomness of the algorithm and* $\mathbb{E}\left[Q_g - g(\mathsf{S} \cup \mathsf{T})\right] \leq 5\Delta/6$ *over the randomness of the algorithm and realizations of* S *and* T*. Here* OPT *is an optimal fully-adaptive algorithm for the stochastic submodular cover problem with the function* g[6]*.*

The goal in Theorem 7.6.2, is to select a set of items that can decrease the deficit of a *typical* realization S of S (i.e., the expected deficit). In order to do so, we first design a non-adaptive algorithm that finds a low-cost set that can decrease the deficit of a *particular* realization S of S. This step is closely related to solving a stochastic submodular maximization problem subject to a knapsack constraint. Indeed, when costs of all the items are the same, i.e., when we want to minimize the number of items in the solution, one can use the algorithm of Asadpour et al. (2008) (with some small modification) for stochastic submodular maximization subject to cardinality constraint for this purpose. Also, when the random

[6] Throughout this chapter we will abuse notation by refering to an optimal fully-adaptive algorithm for different problem instances using the same notation OPT. The specific problem instance will be clear from context.

variables X_i's have binary realizations, i.e. take only two possible values, then one can use the algorithm of (Gupta et al., 2017) for this purpose. However, we are not aware of a solution for the knapsack constraint of the problem in its general form with the bounds required in our algorithms, and hence we present an algorithm for this task as well. The *main step* of our argument is however on how to use this algorithm to prove Theorem 7.6.2, i.e., move from per-realization guarantee, to the expectation guarantee.

7.6.1 A Non-Adaptive Algorithm for Increasing Expected Coverage

We start by presenting a non-adaptive algorithm that picks a low-cost (compared to the *expected* cost of OPT) set of items *deterministically*, while achieving a constant factor of *coverage* of OPT. For any set $A \subseteq [m]$, i.e., the set of indices of stochastic items, and any realization X of X, we define $X_A := \{x_i \mid i \in A\}$, i.e, the realization of all items corresponding to indices in A.

Theorem 7.6.3. *There exists a non-adaptive algorithm that takes as input a set of items $\mathcal{X}$, a monotone submodular function f, and a parameter $\mathbf{Q}$ such that $f(X) = \mathbf{Q}$ for any realization X of $\mathcal{X}$, and outputs a set $A \subseteq [m]$ such that (i) $\mathsf{cost}(\mathsf{X}_A) \leq 3 \cdot \mathbb{E}\left[\mathsf{cost}(\text{OPT})\right]$ and (ii) $\mathbb{E}_{X_A \sim \mathsf{X}}\left[f(X_A)\right] \geq \mathbf{Q}/3$. Here,* OPT *is the optimum adaptive algorithm for submodular cover on $\mathcal{X}$ with function f and parameter $Q = \mathbf{Q}$.*

As argued before, Theorem 7.6.3 can be interpreted as an algorithm for submodular maximization subject to knapsack constraint.

To prove Theorem 7.6.3, we design a simple greedy algorithm (similar to the greedy algorithm for submodular maximization) and analyze it using a linear programming (LP) relaxation in the spirit of Wolsey's LP (Wolsey, 1982) defined in the following section.

Extension of Wolsey's LP for Stochastic Submodular Cover

Let us define the function $F : 2^{[m]} \rightarrow \mathbb{N}_+$ as follows: for any $A \subseteq [m]$,

$$F(A) := \mathop{\mathbb{E}}_{X_A \sim \mathsf{X}}\left[f(X_A)\right]. \tag{7.6.1}$$

As we assume in the lemma statement that $\mathbf{Q} := \mathbb{E}_{X\sim\mathcal{X}}[f(X)]$, we have $F([m]) = \mathbf{Q}$ as well. For any $B \subseteq [m]$, we further define the marginal contribution function $F_B : 2^{[m]} \rightarrow \mathbb{N}_+$ where $F_B(A) := F(A \cup B) - F(B)$ for all $A \subseteq [m] \setminus B$. The following proposition is straightforward.

Proposition 7.6.4. *Function F is a monotone submodular function.*

Proof. F is a convex combination of submodular functions, one for each realization of $\mathcal{X}$.

$\blacksquare$

We will use a linear programming (LP) relaxation in the spirit of Wolsey's LP (Wolsey, 1982) for the submodular cover problem (when applied to the function F). Consider the following linear programming relaxation:

$$\begin{aligned} P = \min_{y\in[0,1]^m} & \sum_{i=1}^{m} c_i \cdot y_i \\ \text{s.t.} & \sum_{i\in[m]\setminus A} F_A(i)\cdot y_i \geq \mathbf{Q} - 2F(A), \quad \forall A \subseteq [m] \end{aligned} \tag{7.6.2}$$

The difference between LP (7.6.2) and Wolsey's LP is in RHS of the constraint which is $\mathbf{Q} - F(A)$ in case Wolsey's LP. In the non-stochastic setting, one can prove that Wolsey's LP lower bounds the value of optimum submodular cover for function F. To extend this result to the stochastic case (for the function f) however, it suffices to modify the constraint as in LP (7.6.2), as we prove in the following lemma.

Lemma 7.6.5. *The cost of an optimal adaptive algorithm* OPT *for submodular cover on function f is lower bounded by the optimal cost P of LP (7.6.2), i.e. $P \leq \mathbb{E}\left[\mathsf{cost}(\text{OPT})\right]$.*

Proof. For a realization X of $\mathcal{X}$ and any $i \in [m]$, define an indicator random variable $w_i(X)$ that takes value 1 iff OPT chooses $\mathcal{X}_i$ on the realization X, i.e.

$$w_i(X) = \mathbf{1}[\mathcal{X}_i \in \text{OPT}(X)].$$

Let w_i be the probability that $\textsc{opt}$ chooses $\mathcal{X}_i$, i.e.,

$$w_i = \Pr_{X \sim \mathsf{X}} \left(w_i(X) = 1 \right) = \mathbb{E}\left[w_i(X) \right].$$

We have that,

$$\mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right] = \mathop{\mathbb{E}}_{X}\left[\sum_{i=1}^{m} \mathbf{1}[\mathcal{X}_i \in \textsc{opt}(X)] \cdot c_i \right] = \sum_{i=1}^{m} w_i \cdot c_i \,.$$

In the following, we prove that $w := (w_1, \ldots, w_m)$ is a feasible solution to LP (7.6.2), which by above equation would immediately imply that $P \leq \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$.

Clearly $w \in [0,1]^m$, so it suffices to prove that the constraint holds for any set $A \subseteq [m]$. The main step in doing so is the following claim.

Claim 7.6.6. *For any set $A \subseteq [m]$, and any two realizations X and X' of X:*

$$f(X_A) + f(X'_A) + \sum_{i \in [m] \setminus A} f_{X'_A}(x_i) \cdot w_i(X) \geq \mathbf{Q}.$$

Proof. Recall that we assume $f(X) = \mathbf{Q}$ always, and hence $f(\textsc{opt}(X)) = \mathbf{Q}$ as well. Moreover, for any $i \in \textsc{opt}(X)$, $w_i(X) = 1$ and for $i \in [m] \setminus \textsc{opt}(X)$, $w_i(X) = 0$. We further define the sets:

$$B := \textsc{opt}(X) \cap A \qquad \text{and} \qquad C := \textsc{opt}(X) \setminus B.$$

We have,

$$\begin{aligned}
f(X_A) + f(X'_A) + \sum_{i \in [m] \setminus A} f_{X'_A}(x_i) \cdot w_i(X) &= f(X_A) + f(X'_A) + \sum_{x_i \in C} f_{X'_A}(x_i) \\
&\underset{\text{Fact 7.4.1}}{\geq} f(X_A) + f(X'_A \cup C) \quad \text{(by submodularity)} \\
&\geq f(X_B) + f(X_C) \\
&\qquad \text{(by monotonicity as } X_B \subseteq X_A) \\
&= f(X_B \cup X_C) = \mathbf{Q}, \\
&\qquad \text{(by submodularity and since } X_B \cup X_C = \textsc{opt}(X))
\end{aligned}$$

which finalizes the proof. ▮ Claim 7.6.6

Fix any set $A \subseteq [m]$. We first take an expectation over all realizations of X in LHS of Claim 7.6.6:

$$\begin{aligned}
\mathbf{Q} &\underset{\text{Claim 7.6.6}}{\leq} \mathop{\mathbb{E}}_{X}\left[f(X_A) + f(X'_A) + \sum_{i \in [m] \setminus A} f_{X'_A}(x_i) \cdot w_i(X)\right] \\
&= \mathop{\mathbb{E}}_{X}\left[f(X_A)\right] + f(X'_A) + \sum_{i \in [m] \setminus A} \mathop{\mathbb{E}}_{X}\left[f_{X'_A}(x_i) \cdot w_i(X)\right] \\
&= \mathop{\mathbb{E}}_{X}\left[f(X_A)\right] + f(X'_A) + \sum_{i \in [m] \setminus A} \mathop{\mathbb{E}}_{X}\left[f_{X'_A}(x_i)\right] \cdot \mathop{\mathbb{E}}_{X}\left[w_i(X)\right],
\end{aligned}$$

as random variables $f_{X'_A}(\mathsf{X}_i)$ and $w_i(\mathsf{X})$ are independent since the choice of X_i by $\textsc{opt}$ is independent of what X_i realizes to. We further point out that $\mathbb{E}_X\left[f(X_A)\right]$ in the RHS of last equation above is equal to $F(A)$ by definition in Eq (7.6.1) and $\mathbb{E}_X\left[w_i(X)\right] = w_i$.

We further take an expectation over all realizations of X' in the RHS above:

$$\begin{aligned}
\mathbf{Q} &\leq \mathop{\mathbb{E}}_{X'}\left[F(A) + f(X'_A) + \sum_{i\in[m]\setminus A} \mathop{\mathbb{E}}_{X}\left[f_{X'_A}(x_i)\right]\cdot w_i\right] \\
&\underset{\text{Eq (7.6.1)}}{=} F(A) + F(A) + \sum_{i\in[m]\setminus A} \mathop{\mathbb{E}}_{X'}\mathop{\mathbb{E}}_{X}\left[f_{X'_A}(x_i)\right]\cdot w_i \\
&= 2\cdot F(A) + \sum_{i\in[m]\setminus A} F_A(i)\cdot w_i\,,
\end{aligned}$$

as $F_A(i) = \mathbb{E}_{X'}\,\mathbb{E}_X\left[f(X'_A \cup X_i) - f(X'_A)\right]$. Rewriting the above equation, we obtain that the constraint associated with set A is satisfied by w. This concludes the proof that w is a feasible solution. ■ Lemma 7.6.5

The Non-Adaptive-Greedy Algorithm

We now design an algorithm, namely NON-ADAPT-GREEDY, based on "the greedy algorithm" (for submodular optimization) applied to the function F in the last section and then use LP (7.6.2) to analyze it. We emphasize that the use of the LP is only in the analysis and not in the algorithm.

NON-ADAPT-GREEDY$(\mathsf{X}, f, \mathbf{Q})$. Given a monotone submodular function f, the set of stochastic items X, and a parameter $\mathbf{Q} = f(X)$ for all X, outputs a set A of (indices of) stochastic items.

```
1. Initialize: Set $A \leftarrow \emptyset$ and $F$ be the function associated to $f$ in Eq (7.6.1).
2. While $F(A) < \mathbf{Q}/3$ do:
    (a) Let $j^* \leftarrow \text{argmax}_{j\in[m]} F_A(j)/c_j$.
    (b) Update $A \leftarrow A \cup \{j^*\}$.
3. Output: $A$.
```

It is clear that the set A output by NON-ADAPT-GREEDY achieves $\mathbb{E}_{X_A}\left[f(X_A)\right] = F(A) \geq \mathbf{Q}/3$ (as $F([m]) = \mathbf{Q}$, the termination condition would always be satisfied eventually). We will now bound the cost paid by the greedy algorithm in terms of the optimal value P of

LP (7.6.2).

Lemma 7.6.7. $\mathsf{cost}(\mathsf{X}_A) \leq 3P$.

To prove Lemma 7.6.7 we need some definition. Let the sequence of items picked by the greedy algorithm be $j_1, j_2, j_3 \cdots$, where j_i is the index of the item picked in iteration i. Moreover, for any i, define $A_{<i} := \{j_1, \ldots, j_{i-1}\}$, i.e., the set of items chosen before iteration i. We first prove the following bound on the ratio of coverage rate to costs in each iteration.

Lemma 7.6.8. *In each iteration i of the non-adaptive greedy algorithm we have,*

$$\frac{F_{A_{<i}}(j_i)}{c_{j_i}} \geq \frac{\mathbf{Q} - 2F(A_{<i})}{P},$$

where P is the optimal value of LP (7.6.2).

Proof. Fix any iteration i. Recall that in each iteration, we pick item $j_i \in \text{argmax}_{j\in[m]} F_{\mathbf{A}_{<i}}(j)/c_j$. Suppose towards a contradiction that in some iteration i:

$$\forall j \in [m] \qquad \frac{F_{\mathbf{A}_{<i}}(j)}{c_j} < \frac{\mathbf{Q} - 2F(A_{<i})}{P}. \tag{7.6.3}$$

Let y^* be an optimal solution to LP (7.6.2), then by the constraint of the LP for set $A_{<i}$ we have

$$\begin{aligned}
\mathbf{Q} - 2F(A_{<i}) &\leq \sum_{j\in[m]\setminus A_{<i}} F_{A_{<i}}(j) \cdot y_j^* \\
&\underset{\text{Eq (7.6.3)}}{<} \sum_{j\in[m]\setminus A_{<i}} y_j^* \cdot c_j \cdot \frac{\mathbf{Q} - 2F(A_{<i})}{P} \\
&\leq \frac{\mathbf{Q} - 2F(A_{<i})}{P} \cdot \sum_{j\in[m]} y_j^* c_j = \mathbf{Q} - 2F(A_{<i}),
\end{aligned}$$

where the last equality is because by definition $\sum_{j\in[m]} y_j^* c_j = P$. By above equation, $\mathbf{Q} - 2F(A_{<i}) < \mathbf{Q} - 2F(A_{<i})$, a contradiction. ▮ Lemma 7.6.8

Proof of Lemma 7.6.7. Fix any iteration i in the algorithm where $F(A_{<i}) \leq \mathbf{Q}/3$. By Lemma 7.6.8,

$$F_{A_{<i}}(j_i) \underset{\text{Lemma 7.6.8}}{\geq} c_{j_i} \cdot \frac{\mathbf{Q} - 2F(A_{<i})}{P} \geq c_{j_i} \cdot \frac{\mathbf{Q}}{3P}. \tag{7.6.4}$$

Let k be the first index where $F_{A_{<k}} < \mathbf{Q}/3$ but $F_{A_{<k+1}} \geq \mathbf{Q}/3$ (i.e., the iteration the algorithm terminates). Note that $\mathsf{cost}(\mathsf{X}_A) = \sum_{i=1}^{k} c_{j_i}$. We start by bounding the first $k-1$ terms in $\mathsf{cost}(\mathsf{X}_A)$:

$$\begin{aligned}\mathbf{Q}/3 > F(A_{<k}) = \sum_{i=1}^{k-1} F_{A_{<i}}(j_i) &\underset{\text{Eq (7.6.4)}}{\geq} \sum_{i=1}^{k-1} c_{j_i} \cdot \frac{\mathbf{Q}}{3P} \\ \implies \sum_{i=1}^{k-1} c_{j_i} &< P.\end{aligned}$$

Now consider the last term in $\mathsf{cost}(A)$, i.e., c_{j_k}. Again, by Lemma 7.6.8, we have,

$$c_{j_k} \underset{\text{Lemma 7.6.8}}{\leq} \frac{F_{A_{<k}}(j_k) \cdot P}{\mathbf{Q} - 2F(A_{<k})} \leq \frac{(\mathbf{Q} - F(A_{<k})) \cdot P}{\mathbf{Q} - 2F(A_{<k})} \leq 2P,$$

using the fact that $F(A_{<k}) < \mathbf{Q}/3$. As such, $\mathsf{cost}(\mathsf{X}_A) \leq 3P$ finalizing the proof. ■ Lemma 7.6.7

Theorem 7.6.3 now follows immediately from Lemma 7.6.7 and Lemma 7.6.5 as $P \leq \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$.

7.6.2 Proof of Theorem 7.6.2

We use the algorithm in Theorem 7.6.3 to present the following algorithm for reducing the expected deficit of any given set S in Theorem 7.6.2.

$\textsc{Select}(\mathsf{X}, g, \mathsf{S}, \alpha)$. Given a collection of indices $\mathcal{X}$, a monotone submodular function g with $g(X) = Q_g$ for every $X \sim \mathcal{X}$, collection of items $\mathcal{S}$ with expected deficit $\Delta = \mathbf{E}[Q_g - g(\mathcal{S})]$, picks a set T of items to decrease the expected deficit.

1. Let $\Psi := 6\alpha$.

```
2. For $i = 1, \cdots, \Psi$ do:
    (a) Sample a realization $S_i \sim \mathcal{S}$.
    (b) $\mathsf{T}_i \leftarrow \textsc{non-adapt-greedy}(\mathsf{X} \setminus \mathsf{S}, g_{S_i}, \Delta(S_i))$ (recall that $\Delta(S_i) = Q_g - g(S_i)$).
3. Return all items in the sets $\mathsf{T} := \mathsf{T}_1 \cup \mathsf{T}_2 \cdots \cup \mathsf{T}_\Psi$.
```

The Select algorithm repeatedly calls the non-adapt-greedy algorithm for samples drawn from realizations of the set S. By Fact 7.4.2, for any realization S_i of S, $g_{S_i}(\cdot)$ is also a monotone submodular function. Moreover, by the assumption that $g(X) = Q_g$ always, we have that $g_{S_i}(X \setminus S_i) = Q_g - f(S_i)$ always as well. Hence, the parameters given to function non-adapt-greedy in Select are valid.

We first bound the expected cost of Select.

Claim 7.6.9. $\mathbb{E}\left[\mathsf{cost}(\mathsf{T})\right] = O(\alpha) \cdot \mathbb{E}\left[\mathsf{cost}(\textsc{opt})\right]$.

Proof. Cost of T is the cost of the sets $\mathsf{T}_1, \ldots, \mathsf{T}_\Psi$ chosen by non-adapt-greedy on g_{S_i} for each of the Ψ realizations of S. By Theorem 7.6.3, we can bound the cost of each T_i using opt conditioned on realization S_i for S (as we consider g_{S_i}). As such,

$$\begin{aligned}
\mathbb{E}\left[\mathsf{cost}(\mathsf{T})\right] &= \sum_{i=1}^{\Psi} \mathop{\mathbb{E}}_{S_i \sim \mathsf{S}}\left[\mathsf{cost}(\mathsf{T}_i)\right] \\
&\underset{(a)}{\leq} \sum_{i=1}^{\Psi} \mathop{\mathbb{E}}_{S_i \sim \mathsf{S}}\left[3 \cdot \mathop{\mathbb{E}}_{X}\left[\mathsf{cost}(\textsc{opt}(X)) \mid \mathsf{S} = S_i\right]\right] \\
&= \sum_{i=1}^{\Psi} 3 \cdot \mathop{\mathbb{E}}_{S_i \sim \mathsf{S}} \mathop{\mathbb{E}}_{X \sim \mathsf{X} \mid S_i}\left[\mathsf{cost}(\textsc{opt}(X))\right] \\
&= 3\Psi \cdot \mathop{\mathbb{E}}_{X}\left[\mathsf{cost}(\textsc{opt}(X))\right] ,
\end{aligned}$$

where the inequality (a) follows from Theorem 7.6.3 because even though the opt used in Theorem 7.6.3 is an optimal algorithm on the problem instance $(\widetilde{Q}, \mathcal{X} \setminus \mathcal{S})$, but the cost of $\mathbb{E}_X\left[\mathsf{cost}(\textsc{opt}(X)) \mid \mathsf{S} = S_i\right]$ can only be larger than the cost of opt on the instance $(\widetilde{Q}, \mathcal{X} \setminus \mathcal{S})$.

The bound now follow from the value of $\Psi = 6\alpha$. $\blacksquare$

We now prove that the expected deficit of $f(\mathsf{S} \cup \mathsf{T})$ is dropped by at least a $\Delta/6$ factor. The following lemma is at the heart of the proof.

Lemma 7.6.10. $\mathbb{E}\left[\Delta(\mathsf{S} \cup \mathsf{T})\right] \leq 5\Delta/6$.

Proof. We start by introducing the notation needed in the proof. It is useful to note that the randomness in T_i is due to two sources: (1) the sample $S_i \sim \mathcal{S}$ which determines which sets are *indexed* by T_i; and (2) the randomness in the *realization* of the sets indexed by T_i. For any realization S of S, we use $\mathsf{T}_i(S)$ to denote the set T_i chosen (deterministically now by NON-ADAPT-GREEDY) conditioned on $\mathsf{S} = S$ (this corresponds to "fixing" the first source of randomness above). We use the notation $\mathsf{T}_{\leq i}$ to denote the collection $\mathsf{T}_1 \cup \cdots \cup \mathsf{T}_i$ of sets selected in iterations 1 through i, and $S_{\leq i}$ to denote the tuple of realizations $(S_1, \cdots, S_i)$ (we define $\mathsf{T}_{<i}$ and $S_{<i}$ analogously, where $\mathsf{T}_{<1} = S_{<1} = \emptyset$). We also denote by $\mathsf{T}_{\leq i}(S_{\leq i})$ the sets selected in iterations 1 to i given $S_{\leq i}$.

Consider any $i \in [\Psi]$. For a realization $S_i \sim \mathsf{S}$, we are computing NON-ADAPT-GREEDY on g_{S_i} with parameter $\mathbf{Q} = \Delta(S_i)$. As such, by Theorem 7.6.3, for the set $\mathsf{T}_i(S_i)$ returned, we have $\mathbb{E}_X\left[g_{S_i}(\mathsf{T}_i(S_i))\right] \geq \mathbf{Q}/3 = \Delta(S_i)/3$. Consequently,

$$\underset{S_i \sim \mathcal{S}}{\mathbb{E}}\, \underset{X}{\mathbb{E}}[g_{S_i}\big(\mathsf{T}_i(S_i)\big)] \geq \underset{S_i \sim \mathcal{S}}{\mathbb{E}}[\frac{\Delta(S_i)}{3}] = \frac{\Delta}{3}. \tag{7.6.5}$$

We now use this equation to argue that adding each set T_i can decrease the expected deficit. Before that, let us briefly touch upon the difficulty in proving this statement and the intuition behind the proof. In SELECT, we first pick a realization S_i of S and then add "enough" sets to T_i to (almost) cover the deficit *introduced by* S_i. This corresponds to Eq (7.6.5). However, our goal is to decrease the *expected* deficit of S (not a deficit of a single realization). As such, the quantity of interest is in fact the following instead:

$$\underset{X}{\mathbb{E}}\left[g_{\mathsf{S}}(\mathsf{T}_i)\right] = \underset{S_i\sim\mathsf{S}}{\mathbb{E}}\;\underset{S'_i\sim\mathsf{S}}{\mathbb{E}}\;\underset{X}{\mathbb{E}}\left[g_{S'_i}(\mathsf{T}_i(S_i))\right], \tag{7.6.6}$$

i.e., the marginal contribution of $\mathsf{T}_i(S_i)$ (chosen by picking a set S_i) to a "typical" set $S'_i \sim \mathsf{S}_i$ (not exactly the set S_i). The set T_i we picked in this step is *not necessarily* covering the deficit introduced by S'_i as well (in the context of the stochastic set cover problem, think of S_i and S'_i as covering a completely different set of elements and T_i being a deterministic set covering $U \setminus S_i$). As such, it is not at all clear that picking the set T_i should make "any progress" towards reducing the expected deficit.

The way we get around this difficulty is to additionally consider the marginal contribution of the sets $\mathsf{T}_1, \ldots, \mathsf{T}_\Psi$ to *each other*. If T_1 cannot decrease the expected deficit of most realizations S chosen from S, then this means that by picking another set $\mathsf{T}_2(S)$ (for a realization S of S), the set $\mathsf{T}_1 \cup \mathsf{T}_2$ needs to have a coverage larger than both T_1 and T_2 individually (in the context of the set cover problem, since T_1 is "useless" in covering deficit created by S, and T_2 can cover this deficit, this means that T_1 and T_2 should not have many elements in common typically). We formalize this intuition in the following claim (compare Eq (7.6.7) in this claim with Eq (7.6.6)).

Claim 7.6.11. *Suppose at the start of iteration i the following holds*

$$\underset{S_i\sim\mathcal{S}}{\mathbb{E}}\;\underset{S_{<i}\sim\mathcal{S}}{\mathbb{E}}\;\underset{X}{\mathbb{E}}[g_{S_i}\left(\mathsf{T}_{<i}(S_{<i})\right)] < \frac{\Delta}{6}. \tag{7.6.7}$$

Then,

$$\underset{S_{\leq i}\sim\mathcal{S}}{\mathbb{E}}\;\underset{X}{\mathbb{E}}\left[g_{T_{<i}(S_{<i})}(T_i(S_i))\right] > \frac{\Delta}{6}.$$

Proof. By subtracting Eq. (7.6.7) from Eq. (7.6.5), and using linearity of expectation we get that:

$$\begin{aligned}
\frac{\Delta}{6} &< \mathop{\mathbb{E}}_{S_i \sim \mathcal{S}} \mathop{\mathbb{E}}_{S_{<i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} [g_{S_i}\big(\mathsf{T}_i(S_i)\big) - g_{S_i}\big(\mathsf{T}_{<i}(S_{<i})\big)] \\
&= \mathop{\mathbb{E}}_{S_i \sim \mathcal{S}} \mathop{\mathbb{E}}_{S_{<i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} [g\big(\mathsf{T}_i(S_i) \cup S_i\big) - g\big(\mathsf{T}_{<i}(S_{<i}) \cup S_i\big)] \\
&\leq \mathop{\mathbb{E}}_{S_i \sim \mathcal{S}} \mathop{\mathbb{E}}_{S_{<i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} [g\big(\mathsf{T}_{\leq i}(S_{\leq i}) \cup S_i\big) - g\big(\mathsf{T}_{<i}(S_{<i}) \cup S_i\big)] && \text{(by monotonicity)} \\
&\leq \mathop{\mathbb{E}}_{S_i \sim \mathcal{S}} \mathop{\mathbb{E}}_{S_{<i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} [g\big(\mathsf{T}_{\leq i}(S_{\leq i})\big) - g\big(\mathsf{T}_{<i}(S_{<i})\big)] \\
& && \text{(by submodularity as } T_{<i}(S_{<i}) \subseteq T_{\leq i}(S_{\leq i})) \\
&= \mathop{\mathbb{E}}_{S_{\leq i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} \left[g_{T_{<i}(S_{<i})}(T_i(S_i)) \right], && (7.6.8)
\end{aligned}$$

finalizing the proof. $\blacksquare$ Claim 7.6.11

Suppose towards a contradiction that $\mathbb{E}\left[\Delta(\mathsf{S} \cup \mathsf{T})\right] > 5\Delta/6$. This implies that,

$$\begin{aligned}
5\Delta/6 < \mathbb{E}\left[Q_g - g(\mathsf{S} \cup \mathsf{T})\right] = \mathbb{E}\left[Q_g - g(\mathsf{S}) - g_{\mathsf{S}}(\mathsf{T})\right] \\
\implies \mathop{\mathbb{E}}_{S \sim \mathsf{S}} \mathop{\mathbb{E}}_{X} \left[g_S(\mathsf{T})\right] < \Delta/6.
\end{aligned}$$

By monotonicity of f and since $\mathsf{T} = \mathsf{T}_1 \cup \ldots \cup \mathsf{T}_\Psi$, this implies that for all $i \in [\Psi]$,

$$\Delta/6 > \mathop{\mathbb{E}}_{S \sim \mathsf{S}} \mathop{\mathbb{E}}_{X} \left[g_S(\mathsf{T}_{\leq i})\right] = \mathop{\mathbb{E}}_{S_i \sim \mathcal{S}} \mathop{\mathbb{E}}_{S_{<i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} [g_{S_i}\big(\mathsf{T}_{<i}(S_{<i})\big)].$$

Hence, we can apply Claim 7.6.11 to obtain that for any $i \in [\Psi]$:

$$\mathop{\mathbb{E}}_{S_{\leq i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} \left[g_{T_{<i}(S_{<i})}(T_i(S_i)) \right] > \frac{\Delta}{6}.$$

As such, by linearity of expectation and above equation,

$$\begin{aligned}\mathop{\mathbb{E}}_{S_{\leq \Psi} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} \left[g(\mathsf{T}(S_{\leq \Psi}))\right] &= \sum_{i=1}^{\Psi} \mathop{\mathbb{E}}_{S_{\leq i} \sim \mathcal{S}} \mathop{\mathbb{E}}_{X} \left[g_{T_{<i}(S_{<i})}(\mathsf{T}_i(S_i))\right] \\ &> \Psi \cdot \frac{\Delta}{6} = 6\alpha \cdot \frac{\Delta}{6} \\ &\geq Q_g = \mathop{\mathbb{E}}_{X}[g(\mathcal{X})].\end{aligned}$$

where the last inequality follows due to the condition that $\alpha \geq Q_g/\Delta$. The above is a contradiction as $\mathsf{T} \subseteq \mathcal{X}$ and g is monotone. Hence, $\mathbb{E}\left[\Delta(\mathsf{S} \cup \mathsf{T})\right] \leq 5\Delta/6$, finalizing the proof.

■ Lemma 7.6.10

Theorem 7.6.2 now follows immediately from Claim 7.6.9 and Lemma 7.6.10.

7.7 Algorithms for the Stochastic Submodular Cover Problem

In this section, we present our main algorithmic result which formalizes Result 1.

Theorem 7.7.1. *Let E be a ground-set of items, $f : 2^E \rightarrow \mathbb{N}_+$ be a monotone submodular function with $Q := f(E)$, and $\mathsf{X} := \{\mathsf{X}_1, \ldots, \mathsf{X}_m\}$ be a collection of m stochastic items with support in E. Let $c_i \in [C]$ be the integer-valued cost of item $\mathcal{X}_i$. For any integer $r \geq 1$, there exists an r-round adaptive algorithm for the stochastic submodular cover problem for function f and items $\mathcal{X}$ with expected cost $O(r \cdot Q^{1/r} \cdot \log Q \cdot \log(mC))$ times the expected cost of the optimal adaptive algorithm.*

Theorem 7.7.1 immediately implies that the r-round adaptivity gap of the stochastic submodular cover problem is $\widetilde{O}(Q^{1/r})$. The rest of this section is devoted to the proof of Theorem 7.7.1.

Overview. The underlying strategy behind our algorithm is as follows: in each round of the algorithm, reduce the *deficit* of the currently realized set T chosen in the previous rounds (i.e., the quantity $Q - f(T)$) by a factor of roughly $Q^{1/r}$. This suggests that after r rounds the deficit should reach zero, hence we obtain a submodular cover. In order to do so, the algorithm needs to specify an ordering of items *without* knowing the realizations of these items in advance (i.e., non-adaptively). This step is itself done by running the algorithm in Theorem 7.6.2 over multiple iterative *phases* to reduce the deficit without knowing realization of any chosen items in this round. We now present our algorithm in details, starting with its main component for reducing the deficit in each round.

7.7.1 The Reduce Subroutine

Let T_k be the items selected by the r-round adaptive algorithm in rounds up to (and including) k, and T_k be their realization. In round k, the algorithm creates an ordering of all the available items and sets a threshold $\tau_k := Q - Q^{(r-k)/r}$ for coverage in this round: after deciding on an ordering of the items non-adaptively, the algorithm picks items according to this ordering one by one until the total coverage of the function reaches τ_k. In this section, we design an algorithm, namely Reduce, which returns an *ordered set* $\mathsf{S} \subseteq \mathcal{X} \setminus \mathsf{T}_{k-1}$ in round k such that items in S are enough to reach the coverage threshold for this round with high probability. If there are items that are not included in S by Reduce, we will simply add them at the end of S in any arbitrary order.

The input to the function Reduce in round k is the set of items $\mathsf{X} \setminus \mathsf{T}_{k-1}$, and the function marginal $f_{T_{k-1}}$; by Fact 7.4.2, $f_{T_{k-1}}$ is also a monotone submodular function. The execution of Reduce is partitioned over $\Gamma := O(\log(mC))$ *phases*, where in each phase, the algorithm picks a new set of items to be added to the (ordered) set returned by it. The final set of items returned by Reduce are ordered in increasing order of the phases (with arbitrary ordering in each phase).

For any phase $p \in [\Gamma]$, we define $\mathcal{S}_p$ as the ordered set of items selected in phase 1 up to (and including) p. Let $Q_k := Q - f(T_{k-1})$; this is the *deficit* of the set T_{k-1} with respect to function f. For any set S of items, we define the following event $\mathcal{E}_k(\mathcal{S})$:

$$\mathcal{E}_k(\mathcal{S}) := \mathbf{1}[Q_k - f_{T_{k-1}}(\mathcal{S}) \geq Q_k / Q^{1/r}]. \tag{7.7.1}$$

Intuitively speaking, $\mathcal{E}_k(\mathcal{S})$ happens if the set of items $\mathcal{S}$ cannot cover most of Q_k yet.

In each phase, Reduce makes $\Lambda := O(\log Q)$ calls to Select subroutine (Theorem 7.6.2). Each call in phase p is to increase the coverage of the set $\mathcal{S}_{p-1}$ to eventually achieve a larger coverage in $\mathcal{S}_p$. Instead of passing $\mathcal{S}_{p-1}$ directly to Select, we instead pass the set $\widetilde{\mathsf{S}}_{p-1} := \mathsf{S}_{p-1} \mid \mathcal{E}_k(\mathsf{S}_{p-1})$ which is a set of items that has the same distribution as S_{p-1} conditioned on the event $\mathcal{E}_k(\mathsf{S}_{p-1})$ (i.e., we only consider such realizations of S_{p-1} where $\mathcal{E}_k(\mathcal{S}_{p-1})$ occurs). We show in Claim 7.7.2 that the performance of Select remains the same in this case also (simply because in Select we only access the distribution of input sets by sampling from it and hence we can sample from $\widetilde{\mathsf{S}}_{p-1}$ instead of S_{p-1}). This step is required to ensure that we can indeed achieve a larger coverage with higher probability across phases as we are "focusing" on realizations that are "bad" in previous phases, i.e., cannot cover a large fraction of Q_k. Formally, we prove that the $\Pr(\mathcal{E}_k(\mathsf{S}_p)) \leq 1/2 \cdot \Pr(\mathcal{E}_k(\mathsf{S}_{p-1}))$, hence after $\Gamma = \Theta(\log{(mC)})$ phases, the probability of this bad event reduces to $1/(mC)^{O(1)}$ and we can move on to the next round. We present the pseudo-code of Reduce algorithm below.

```
Reduce($\mathcal{X}, f_{T_{k-1}}$): Given a set $\mathcal{X}$ of items and a monotone submodular function $f_{T_{k-1}}$,
outputs an ordered set of items $\mathcal{S}$ to be used in round $k$ of the $r$-round adaptive algorithm.
  1. Initialize: Set $\Lambda \leftarrow 12\log(Q)$, and $\Gamma \leftarrow 2\log{(mC)}$ .
  2. Set $\mathcal{S}_0 \leftarrow \emptyset$.
  3. For phases $p = 1, \cdots, \Gamma$ do:
     (a) Set $\mathbb{R}_0 \leftarrow \emptyset$ and let $\widetilde{\mathsf{S}}_{p-1} := \mathcal{S}_{p-1} \mid \mathcal{E}_k(\mathcal{S}_{p-1})$.
     (b) For iterations $i = 1, \cdots, \Lambda$ do:
         i. $\mathbb{R}_i \leftarrow \mathbb{R}_{i-1} \cup \textsc{Select}(\mathsf{X} \setminus \{\mathbb{R}_{i-1} \cup \mathcal{S}_{p-1}\}, f_{T_{k-1}}, \mathbb{R}_{i-1} \cup \widetilde{\mathsf{S}}_{p-1}, 2Q^{1/r})$.
     (c) $\mathcal{S}_p \leftarrow \mathcal{S}_{p-1} \cup \mathbb{R}_\Lambda$.
  4. Return the set $\mathcal{S}_\Gamma$, ordered according to the order in which items were added to $\mathcal{S}_\Gamma$.
```

Before analyzing Reduce we need the following straightforward extension of Theorem 7.6.2.

Claim 7.7.2 (Extension of Theorem 7.6.2)**.** *Let f_T be any monotone submodular function,* *for some $T \subseteq E$, such that $Q' := Q - f(T)$. Let $\mathsf{S} \subseteq \mathsf{X}$ be any subset of items, and $\mathcal{E}$ be an event which is a function of $\mathcal{S}$ and $\widetilde{\mathsf{S}} := \mathcal{S} \mid \mathcal{E}$. Let $\Delta := \mathbb{E}[Q' - f_T(\widetilde{\mathsf{S}})]$, then* Select, *given parameter $\alpha \geq Q'/\Delta$, and 6α samples from $\widetilde{\mathsf{S}}$, outputs a set $\mathbb{R} \subseteq \mathsf{X} \setminus \mathsf{S}$ such that cost of $\mathbb{R}$ is $O(\alpha) \cdot \mathbb{E}\left[\mathsf{cost}(\textsc{opt})|\mathcal{E}\right]$ in expectation over the randomness of the algorithm and $\mathbb{E}\left[Q' - f_T(\widetilde{\mathsf{S}} \cup \mathbb{R})\right] \leq 5\Delta/6$ over the randomness of the algorithm and realizations of $\widetilde{\mathsf{S}}$ and $\mathbb{R}$.*

Claim 7.7.2 can be proven as follows: in Select we only need samples from the distribution S, hence by sampling from the distribution of $\widetilde{\mathsf{S}}$ instead we obtain the same result conditioned on event $\mathcal{E}$. One should be careful though, as the items in $\widetilde{\mathsf{S}}$ are no longer independent due to the conditioning on $\mathcal{E}$. However, Select does not require independence between items in $\mathcal{S}$ and we can simply use $\widetilde{\mathsf{S}}$ instead of S.

We start by bounding the cost of the sets returned by Reduce in each phase. Note that not all these sets are going to be chosen by the r-round algorithm in round k (as we may cover τ_k before reaching these sets and move on to next round) and hence this cost is *not* a lower bound on cost of the r-round algorithm.

Claim 7.7.3. *For any $p \in [\Gamma]$, $\mathbb{E}\left[\mathsf{cost}(\mathsf{S}_p \setminus \mathsf{S}_{p-1})\right] = O(Q^{1/r} \cdot \log Q) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|\mathcal{E}_k(\mathcal{S}_{p-1})]$.*

Proof. We call Select with the parameter $2Q^{1/r}$ for $O(\log Q)$ iterations. By Claim 7.7.2, cost of each iteration of phase p is at most $O(Q^{1/r})$ times the expected cost of opt conditioned on $\mathcal{E}_k(\mathcal{S}_{p-1})$. Hence, total cost of phase p is $\mathbb{E}[\mathsf{cost}(\mathsf{S}_p \setminus \mathsf{S}_{p-1})] = O(Q^{1/r} \cdot \log Q) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|\mathcal{E}_k(\mathcal{S}_{p-1})]$. ∎

We now prove the main property of the Reduce subroutine, i.e., that the sets returned by it can cover the required threshold τ_k with high probability.

Lemma 7.7.4. *Suppose $\mathsf{S}_\Gamma :=$ Reduce$(\mathsf{X}, f_{T_{k-1}})$. Then,*

$$\Pr(\mathcal{E}_k(\mathcal{S}_\Gamma)) \leq 1/(mC)^2,$$

with respect to the randomness of the algorithm and the realizations of $\mathcal{S}_\Gamma$.

Proof. We prove that the probability of the event $\mathcal{E}_k(\mathcal{S}_p)$ decreases after each phase p by a constant factor. Fix a phase $p \in [\Gamma]$. For a realization S we define deficit $\Delta(S) = Q_k - f_{T_{k-1}}(S)$. Recall that $\mathbb{R}_i$ is the set of items picked up to (and including) iteration i in phase p on calls to SELECT with parameter $\alpha = 2Q^{1/r}$. By Claim 7.7.2 we know that each iteration reduces the expected deficit by a constant factor. More formally, fix an $\mathbb{R}_{i-1}$ selected up to iteration $i-1$. If $\mathbb{E}\left[\Delta(\mathbb{R}_{i-1} \cup \mathcal{S}_{p-1})|\mathcal{E}_{p-1}\right] \geq Q_k/2Q^{1/r}$, then the condition of Claim 7.7.2 that $\alpha \geq Q'/\Delta$ is satisfied with $\Delta = \mathbb{E}\left[\Delta(\mathbb{R}_{i-1} \cup \mathcal{S}_{p-1})|\mathcal{E}_{p-1}\right]$, $\alpha = 2Q^{1/r}$, and $Q' = Q_k$. We then have

$$\begin{aligned}&\mathbb{E}\left[\Delta(\mathbb{R}_i \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})\right] \\ &\qquad\qquad \leq \frac{5}{6}\,\mathbb{E}\left[\Delta(\mathbb{R}_{i-1} \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})\right],\end{aligned}$$

where the above expectation is also over the randomness of the SELECT subroutine in iteration i, in addition of the realization of $\mathbb{R}_i \cup \mathcal{S}_{p-1}$. Now, we will prove that Λ iterations are enough to drop the expected deficit below $Q_k/2Q^{1/r}$. Suppose for a contradiction that this is not the case, i.e. after Λ iterations we have that

$$\mathbb{E}[\Delta(\mathbb{R}_\Lambda \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})] \geq \frac{Q_k}{2Q^{1/r}}\,. \tag{7.7.2}$$

Due to the fact that $f_{T_{k-1}}$ is a monotone function, we have

$$\mathbb{E}[\Delta(\mathbb{R}_i \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})] \geq \mathbb{E}[\Delta(\mathbb{R}_\Lambda \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})]\,,$$

for all $\mathbb{R}_i$. Then using Eq. (7.7.2) and the above equation, we can observe that the condition of Claim 7.7.2 that $\mathbb{E}\left[\Delta(\mathbb{R}_i \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})\right] \geq Q_k/2Q^{1/r}$ is satisfied for every $\mathbb{R}_i$. This implies that after Λ iterations the expected deficit can be written as

$$\begin{aligned}\mathbb{E}[\Delta(\mathbb{R}_\Lambda \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})] &\le \left(\frac{5}{6}\right)^{\Lambda} \cdot \mathbb{E}[\Delta(\mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})] \\ &\le \left(\frac{5}{6}\right)^{12\log Q} \cdot Q_k \qquad \text{(Recall that } \Lambda = 12\log Q\text{)} \\ &< \frac{Q_k}{2Q} \le \frac{Q_k}{2Q^{1/r}}\,. \qquad (7.7.3)\end{aligned}$$

Eq. (7.7.2) and Eq. (7.7.3) lead to a contradiction. Hence, we will have that

$$\mathbb{E}[\Delta(\mathcal{S}_p)|\mathcal{E}_k(\mathcal{S}_{p-1})] = \mathbb{E}[\Delta(\mathbb{R}_\Lambda \cup \mathcal{S}_{p-1})|\mathcal{E}_k(\mathcal{S}_{p-1})] < \frac{Q_k}{2Q^{1/r}}\,.$$

where again the expectation is over the randomness of the SELECT subroutine. Now, using Markov's inequality we have that

$$\Pr\left(\mathcal{E}_k(\mathcal{S}_p)\,\middle|\,\mathcal{E}_k(\mathcal{S}_{p-1})\right) = \Pr\left(\Delta(\mathcal{S}_p) \ge \frac{Q_k}{Q^{1/r}}\,\middle|\,\mathcal{E}_k(\mathcal{S}_{p-1})\right) \le \frac{1}{2}\,, \qquad (7.7.4)$$

where the above probability is both with respect to the realizations of $\mathcal{S}_p$ and the coins used by the algorithm to select $\mathcal{S}_p$. Now, we have that

$$\begin{aligned}\Pr(\mathcal{E}_k(\mathsf{S}_\Gamma)) &= \Pr(\mathcal{E}_k(\mathsf{S}_1)) \cdot \Pi_{p=2}^{\Gamma} \Pr\left(\mathcal{E}_k(\mathsf{S}_i) \mid \mathcal{E}_k(\mathsf{S}_{i-1})\right) \\ &\underset{\text{Eq (7.7.4)}}{\le} \left(\frac{1}{2}\right)^{\Gamma-1} \le \frac{1}{(mC)^2},\end{aligned}$$

by the choice of $\Gamma = \Theta(\log(mC))$, which proves the desired result. $\blacksquare$ Lemma 7.7.4

7.7.2 The r-Round Adaptive Algorithm

We are now ready to present our r-round algorithm which is based on successive applications of the REDUCE subroutine.

```
$r$-Round-Adaptive($\mathcal{X}, f, Q$): Given a set of items $\mathcal{X}$, a monotone submodular function
$f$, and the desired coverage value $Q$, outputs a set $\mathsf{T}$ such that its realization $T$ is feasible.
  1. Initialize: Set $\mathsf{T}_0 \leftarrow \emptyset, T_0 \leftarrow \emptyset$
  2. For $k = 1, 2, \cdots, r$ do:
      (a) Set threshold $\tau_k \leftarrow Q - Q^{(r-k)/r}$
      (b) $\mathsf{T} \leftarrow \textsc{Reduce}(\mathcal{X} \setminus \mathsf{T}_{k-1}, f_{T_{k-1}})$
      (c) Add the remaining items $\mathcal{X} \setminus (\mathsf{T} \cup \mathsf{T}_{k-1})$ at the end of $\mathsf{T}$ in any arbitrary order.
      (d) Observe the realizations $T'$ of the set of items $\mathsf{T}' \subseteq \mathsf{T}$ selected by running through
          the ordered set $\mathsf{T}$ until a total coverage of $\tau_k$ is reached, i.e. $f(T_{k-1} \cup T') \geq \tau_k$
      (e) $\mathsf{T}_k \leftarrow \mathsf{T}' \cup \mathsf{T}_{k-1}$ and $T_k \leftarrow T' \cup T_{k-1}$
  3. Return $\mathsf{T}_r$ with realization $T_r$ as the final answer.
```

We are now ready to prove Theorem 7.7.1 by analyzing the above algorithm. The overall plan is to bound the cost of each round of the r-round algorithm. In each round the algorithm selects an ordering returned by a call to Reduce and adds the remaining items at the end of this ordering. As argued earlier, not all the sets in the ordering are going to be chosen by the r-round algorithm in round k. We will use Claim 7.7.3 and Lemma 7.7.4 to bound the expected cost of the items selected from the ordering in round k in terms of the expected cost of OPT. In order to do so, we first lower bound the cost of OPT.

Claim 7.7.5. *For any (possibly randomly chosen) collection $\mathcal{S} \subseteq \mathcal{X}$, and any event $\mathcal{E}$ which is a function of $\mathcal{S}$, the expected cost of* OPT *can be lower bounded as*

$$\mathbb{E}[\mathsf{cost}(\textsc{opt})] \geq \Pr(\mathcal{E}) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt}) | \mathcal{E}].$$

Proof. The expected cost of OPT can be written as

$$\begin{aligned}\mathbb{E}[\mathsf{cost}(\textsc{opt})] &= \Pr(\mathcal{E}) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|\mathcal{E}] + \Pr(\neg\mathcal{E}) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|\neg\mathcal{E}] \\ &\geq \Pr(\mathcal{E}) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|\mathcal{E}] \,.\end{aligned}$$

Note that the above also holds even if the collection $\mathcal{S}$ is itself randomly chosen. ▮ Lemma 7.7.5

We now prove the lemma bounding the expected cost of each round of r-ROUND-ADAPTIVE. We will define the notation $\mathsf{cost}(\textsc{Round}_k)$ to be the total cost of all the items added to the feasible set in round k. More formally,

$$\mathsf{cost}(\textsc{Round}_k) := \mathsf{cost}(\mathsf{T}_k \setminus \mathsf{T}_{k-1}) \,.$$

Now, we will provide a bound on $\mathbb{E}[\mathsf{cost}(\textsc{Round}_k)]$.

Lemma 7.7.6. *For any round $k \leq r$, given T_{k-1}, the expected cost paid by the r-ROUND-ADAPTIVE algorithm in round k can bounded as*

$$\mathbb{E}[\mathsf{cost}(\textsc{Round}_k)|T_{k-1}] \leq O(Q^{1/r} \log(Q) \log(mC)) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|T_{k-1}] \,.$$

Proof. Recall that in round k we call REDUCE with parameter $f_{T_{k-1}} = f_{T_{k-1}}$ such that $Q_k = Q - f(T_{k-1})$. Also, recall that in phase p, REDUCE adds items $\mathcal{S}_p \setminus \mathcal{S}_{p-1}$ to the ordering $\mathcal{S}_\Gamma$ returned by it. Using Claim 7.7.3 we have that

$$\mathbb{E}[\mathsf{cost}(\mathcal{S}_p \setminus \mathcal{S}_{p-1})|T_{k-1}] = O(Q^{1/r} \cdot \log Q) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|T_{k-1}, \mathcal{E}_k(\mathcal{S}_{p-1})] \,.$$

Also, recall that while running through the ordered set of round k we select items from $\mathcal{S}_p \setminus \mathcal{S}_{p-1}$ only if the realization is such that the items in $\mathcal{S}_{p-1}$ are not able to reach the required coverage threshold τ_k. More formally, we only pay for the cost of items in $\mathcal{S}_p \setminus \mathcal{S}_{p-1}$ when the event $\mathcal{E}_k(\mathcal{S}_{p-1})$ occurs. Hence, we will pay the cost of phase p items with probability $\Pr(\mathcal{E}_k(\mathcal{S}_{p-1}))$.

Also, in the case that all the items $\mathcal{S}_\Gamma$ returned by REDUCE are not able to reach the required coverage threshold, we trivially bound the cost by mC. Since, $Q_k \le Q^{(r-k+1)/r}$, this event happens with probability at most $\Pr(\mathcal{E}_k(\mathcal{S}_\Gamma))$ which is upper bounded by $1/(mC)^2$ using Lemma 7.7.4. Combining all this, we have that, given T_{k-1},

$$\begin{aligned}
&\mathbb{E}[\mathsf{cost}(\textsc{Round}_k)|T_{k-1}] \\
&\quad \le \sum_{p=1}^{\Gamma} \Pr\left(\mathcal{E}_k(\mathcal{S}_{p-1})\right) \cdot \mathbb{E}[\mathsf{cost}(\mathcal{S}_p \setminus \mathcal{S}_{p-1})|T_{k-1}] + \Pr\left(\mathcal{E}_k(\mathcal{S}_\Gamma)\right) \cdot mC \\
&\quad \underset{\text{Claim 7.7.3}}{\le} \sum_{p=1}^{\Gamma} \Pr\left(\mathcal{E}_k(\mathcal{S}_{p-1})\right) \cdot O\left(Q^{1/r}\log(Q)\right) \cdot \mathbb{E}[\mathsf{cost}(\textsc{opt})|T_{k-1}, \mathcal{E}_k(\mathcal{S}_{p-1})] \\
&\qquad\quad + \Pr\left(\mathcal{E}_k(\mathcal{S}_\Gamma)\right) \cdot mC \\
&\quad \underset{\text{Claim 7.7.5}}{\le} O\left(Q^{1/r}\log(Q)\log(mC)\right) \mathbb{E}\left[\mathsf{cost}(\textsc{opt})|T_{k-1}\right] + \Pr\left(\mathcal{E}_k(\mathcal{S}_\Gamma)\right) \cdot mC \\
&\quad \underset{\text{Lemma 7.7.4}}{\le} O\left(Q^{1/r}\log(Q)\log(mC)\right) \mathbb{E}\left[\mathsf{cost}(\textsc{opt})|T_{k-1}\right] + \frac{1}{(mC)^2} \cdot mC \\
&\quad = O\left(Q^{1/r}\log(Q)\log(mC)\right) \mathbb{E}\left[\mathsf{cost}(\textsc{opt})|T_{k-1}\right] .
\end{aligned}$$

▮ Lemma 7.7.6

We are now ready to prove Theorem 7.7.1 which uses the above lemma to give a combined bound on the cost of all the rounds.

Proof. (of Theorem 7.7.1) We will first divide the $\mathsf{cost}(r\text{-}\textsc{Round-Adaptive})$ into the cost of each round.

$$\mathbb{E}[\mathsf{cost}(r\text{-}\textsc{Round-Adaptive})] = \sum_{k=1}^{r} \mathbb{E}[\mathsf{cost}(\textsc{Round}_k)] \,, \tag{7.7.5}$$

where recall that $\mathsf{cost}(\textsc{Round}_k) := \mathsf{cost}(\mathsf{T}_k \setminus \mathsf{T}_{k-1})$ and T_k are the items selected up to (and including) round k. Let T_k be the realization of T_k. We first need to understand that there are two sources of randomness– 1) due to the coins used by the algorithm to sample the realizations; 2) due to stochastic nature of items. We will first fix the randomness due to the coins used by the algorithm for sampling. Once we fix the realization of coins used by the algorithm, the only randomness in the algorithm is due to the stochastic nature of items. Then for any $k \leq r$, given a fixed realization of coins in rounds up to $k-1$, we have

$$\begin{aligned}\mathbb{E}[\mathsf{cost}(\textsc{Round}_k)] &\underset{\text{Lemma 7.7.6}}{\leq} O\left(Q^{1/r}\log(Q)\log(mC)\right) \cdot \underset{T_{k-1}\sim\mathsf{T}_{k-1}}{\mathbb{E}} \mathbb{E}[\mathsf{cost}(\textsc{opt})|T_{k-1}] \\ &= O\left(Q^{1/r}\log(Q)\log(mC)\right)\mathbb{E}[\mathsf{cost}(\textsc{opt})]\,,\end{aligned}$$

where the last equality is due to the fact that once we fix the randomness due to coins up to round $k-1$, then the realizations T_{k-1} form a partition over the space of all realizations X. Since the choice of coins was arbitrary, we have that $\mathbb{E}[\mathsf{cost}(\textsc{Round}_k)] \leq \widetilde{O}(Q^{1/r})\mathsf{cost}(\textsc{opt})$.

Then, using Eq. (7.7.5) and the above, the total cost can be bounded as

$$\mathsf{cost}(r\text{-}\textsc{Round-Adaptive}) = O\left(rQ^{1/r}\log(Q)\log(mC)\right)\mathbb{E}[\mathsf{cost}(\textsc{opt})]\,.$$

■ Theorem 7.7.1

Remark 7.7.7. *We can implement the r-round algorithm in polynomial time as long as the costs are polynomially bounded, i.e., achieve a pseudo-polynomial time algorithm. Indeed, the only "time consuming" step of the algorithm is to sample from the conditional distribution $\mathcal{S}|\mathcal{E}$ for some event $\mathcal{E}$. This is however is only needed as long as the $\Pr(\mathcal{E}) \geq 1/(mC)^{\Theta(1)}$. Hence, one can use rejection sampling with the total running time bounded by $\mathrm{poly}(QmC)$ to implement this step. The probability that we do not get the required number of samples from the event $\mathcal{E}$ with $\Pr(\mathcal{E}) \geq 1/mC$ after $\mathrm{poly}(QmC)$ trials is negligible, and we can pay for the cost in case this bad event happens.*

7.8 A Lower Bound for r-Round Adaptive Algorithms

In this section, we prove a lower bound on the approximation ratio of any r-round adaptive algorithm for the submodular cover problem and formalize Result 2. We prove this lower bound for the stochastic set cover problem (see Example 7.2.1) which is a special case of the stochastic submodular cover problem.

Theorem 7.8.1. *For any integer $r \geq 1$, any r-round adaptive algorithm for the stochastic set cover problem on instances with m stochastic sets from a universe of size n elements such that $m = n^{O(r)}$ has expected cost $\Omega(\frac{1}{r^3} \cdot n^{1/r})$ times the cost of the optimal adaptive algorithm.*

Theorem 7.8.1 formalizes Result 2 as by definition, $Q = n$ in the stochastic set cover problem.

Overview. Consider first an instance of the stochastic set cover problem which was used in Goemans and Vondrák (2006) for proving a 1-round adaptivity gap. There exists a single stochastic set, say T, which realizes to $U \setminus \{e^\star\}$ for $e^\star$ chosen uniformly at random from U (support of T has n sets). The remaining sets in this instance are n singleton sets that each deterministically realize to some unique element $e \in U$. Solving such an instance adaptively with just two sets, and indeed even in two rounds of adaptivity, is trivial: choose the set T and observe its realization in the first round; next choose the singleton set that covers $e^\star$. However, consider any non-adaptive algorithm for this problem: even though it is obvious that the set T needs to be the first set in the ordering returned by the algorithm, there is no "good" choice for the ordering of the remaining sets as the algorithm is oblivious to the identity of $e^\star$ at this point. It is then fairly easy to see that no matter what ordering the non-adaptive algorithm chooses, in expectation $\Omega(n)$ sets needs to be picked before it could cover $e^\star$ and hence the universe U. An adaptivity gap of $\Omega(n)$ now follows easily from this argument.

Our main contribution in this section is to design a family of instances in this spirit that allows us to extend the above argument to r-round adaptive algorithms. Roughly speaking, these instances are constructed in a way that at the beginning of each round, the algorithm has access to a set that covers a "large" portion of the remaining universe "randomly", but since the realization of this set is not known to the algorithm, unless it picks many more sets, it would not be able to also cover the "remainder of universe" (left out by the realization of the aforementioned set). Morally speaking, this corresponds to replacing the set $\{e^\star\}$ with larger subsets of U in the above argument and then recurse on each subset individually.

The rest of this section is devoted to the proof of Theorem 7.8.1. We start by introducing an algebraic construction of a set-system, named an *edifice*, due to Chakrabarti and Wirth (Chakrabarti and Wirth, 2016) and use it to introduce a family of "hard" instances for the stochastic set cover problem. We then prove that any algorithm with limited rounds of adaptivity on these instances necessarily incurs a large cost compared to the optimal adaptive algorithm and prove Theorem 7.8.1.

Edifice Set-System

An edifice over a universe U of n items is a collection of sets in which for any two sets, either one of them is a subset of the other, or the two sets have a small intersection. Formally:

Definition 7.8.2 (Edifice Set-System (Chakrabarti and Wirth, 2016))**.** *For integers $k \leq s \leq b \leq d$, a (s, b, k, d)-edifice $\mathcal{T}$ over a universe U is a* complete d-ary k-level rooted tree *together with a collection of associated sets, satisfying the following properties:*

1. *Each vertex v in $\mathcal{T}$ is associated with a set $U_v \subseteq U$ such that the set associated to the root of $\mathcal{T}$ is U, and $U_u \subseteq U_v$ if u is a* child *of v in $\mathcal{T}$.*

2. *If v is a leaf of $\mathcal{T}$, then $|U_v| = b$.*

3. *For each leaf u and each node v* not an ancestor *of u in $\mathcal{T}$, $|U_u \cap U_v| \leq s$.*

In this definition, we say that root is at level 1 *of the tree and the leaf-vertices are at level k*

Edifices are typically interesting when the parameter s is small and parameter b is large compared to the size of the universe, i.e., when we have large sets which are almost disjoint from each other in a recursive manner suggested by the tree-structure of an edifice. For our purpose, we are interested in edifices with parameters $r = k \approx s$ (r is the number of rounds we want to prove the lower bound for), $b \approx n^{1/k}$, and $d = n^{O(1)}$ (n is the number of elements in the universe). The existence of such edifices follows from the results in Chakrabarti and Wirth (2016) (see Theorem 3.5; see also RND-set systems in Assadi and Khanna (2018) for a similar construction), which we summarize in the following proposition.

Proposition 7.8.3 (Chakrabarti and Wirth (2016))**.** *For infinitely many integers N and any integer $k \geq 1$, there exists a $(4k, N, k, N^2)$-edifice over a universe U of size N^k.*

Hard Instances for Stochastic Set Cover

Fix an integer $k \geq 1$ and a sufficiently large integer $N \geq k$ and let U be a universe of size N^k elements. Define $\mathcal{T}$ as any arbitrary $(4k, N, k, N^2)$-edifice over U which is guaranteed to exist by Proposition 7.8.3. We define the following family of "hard" instances for stochastic set cover.

Family $\mathsf{X}^{(k)}$: A collection of stochastic sets over universe U using edifice $\mathcal{T}$.

- For any vertex $u \in \mathcal{T}$ and any element $e \in U$, there exists a dedicated stochastic set X_u and X_e in $\mathsf{X}^{(k)}$, respectively, defined as follows.
- For any non-leaf vertex $u \in \mathcal{T}$ with child-vertices $v_1, \ldots, v_d$, the stochastic set X_u realizes to one of the sets $T_{u,v_1}, \ldots, T_{u,v_d}$ uniformly at random where $T_{u,v_i} := U_u \setminus U_{v_i}$.
- For any leaf vertex $u \in \mathcal{T}$ with $U_u = \{e_1, \ldots, e_N\}$ (recall that $|U_u| = N$ be Definition 7.8.2), the stochastic set X_u realizes to one of the sets $T_{u,e_1}, \ldots, T_{u,e_N}$ uniformly at random where $T_{u,e_i} := U_u \setminus \{e_i\}$.
- For any element $e \in U$, X_e deterministically realizes to the singleton set $\{e\}$.

For any realization of $\mathsf{X}^{(k)}$, we define the *canonical path* of the realization as the root-to-leaf path $P = v_1, v_2, \ldots, v_k$ over the vertices of the edifice $\mathcal{T}$ as follows:

1. v_1 is the root of the tree $\mathcal{T}$.

2. For any $1 < i \leq k$, v_i is the child-vertex of v_{i-1} corresponding to $T_{v_{i-1},v_i} = \mathsf{X}_{v_{i-1}}$.

We have the following simple claim on the cost of the optimal adaptive algorithm on the family $\mathsf{X}^{(k)}$ for any integer $k \geq 1$.

Claim 7.8.4. *For any integer $k \geq 1$, the expected cost of* OPT *on $\mathsf{X}^{(k)}$ is at most $k + 1$.*

Proof. We prove that the following algorithm has expected cost $k+1$; clearly optimal adaptive algorithm can only have a lower expected cost.

Consider the adaptive algorithm that constructs the canonical path of the underlying realization one vertex at a time: it first chooses v_1 which is the root of $\mathcal{T}$ and add X_{v_1} to S. Next, based on the realization of X_{v_1}, it can determine the second vertex v_2 in the canonical path and adds X_{v_2} to S. It continues like this until it has added all sets $\mathsf{X}_{v_1}, \ldots, \mathsf{X}_{v_k}$ to S where $P := v_1, \ldots, v_k$ is the canonical path of the realization. Finally, a realization of X_{v_k} for a leaf v_k corresponds to a set $T_{v_k,e}$ that covers all of U_{v_k} (the set associated with the leaf-vertex v_k in the edifice) except for a single element e. The algorithm then picks the set X_e which deterministically realizes to $\{e\}$.

Clearly, the number of stochastic sets picked by this algorithm is $k+1$. We argue that these sets cover the universe U entirely. This is because, X_{v_1} covers $U \setminus U_{v_2}$, X_{v_2} covers $U_{v_2} \setminus U_{v_3}$, and so on until X_{v_k} covers $U_{v_k} \setminus \{e\}$. As such, $\mathsf{X}_{v_1} \cup \ldots \cup \mathsf{X}_{v_k}$ covers $U \setminus \{e\}$ and picking X_e would cover the whole universe as X_e always realizes to $\{e\}$. ∎

In the remainder of this section, we prove that any $(r =)k$-round adaptive algorithm for stochastic set cover on $\mathsf{X}^{(k)}$ should incur a cost of roughly $n^{1/k}$, hence proving Theorem 7.8.1. It is worth remarking that the adaptive algorithm in Claim 7.8.4 that achieves the cost of $k+1$ requires only $k+1$ rounds of adaptivity; as such, our results are in fact proving a separation between the cost of any k-round and $k+1$-round adaptive algorithms.

Before we move on to the proof of Theorem 7.8.1, we prove the following crucial lemma using properties of edifice $\mathcal{T}$.

Lemma 7.8.5. *Let U_{v_k} be the set associated to the k-th vertex v_k in the canonical path of $\mathsf{X}^{(k)}$ in edifice $\mathcal{T}$ and C be any collection of sets in $\mathsf{X}^{(k)} \setminus \mathsf{X}_{v_k}$. Then $\left|\bigcup_{T\in C} T \cap U_{v_k}\right| \leq 4\,|C| \cdot k$.*

Proof. Fix any set $T \in C$. We prove that $|T \cap U_{v_k}| \leq 4k$ which would immediately imply the lemma.

If T is a realization of some set X_e for some element $e \in U$, then $|T| = 1$ and hence the claim immediately holds. Hence, suppose that T is a realization of X_v for some vertex $v \in \mathcal{T}$.

If v is an ancestor of v_k, then $T = U_v \setminus U'_v$ where v' is either another ancestor of v_k or it is equal to v_k itself by definition of the canonical path. In either case, by property (I) of edifices in Definition 7.8.2, $U_{v_k} \subseteq U_{v'}$ and hence $T \cap U_{v_k} = \emptyset$.

If v is not an ancestor of v_k, then $T \subseteq U_v$ as $\mathsf{X}_v \subseteq U_v$ and by property (III) of edifices in Definition 7.8.2, $|U_v \cap V_{v_k}| \leq 4k$ (here parameter $s = 4k$) and hence $|T \cap V_{v_k}| \leq 4k$, finalizing the proof. ∎

Proof of Theorem 7.8.1

Fix any $k \geq 1$ and a k-round algorithm $\mathcal{A}$ for the stochastic set cover problem on instance $\mathsf{X}^{(k)}$. By Yao's minimax principle (Yao, 1979), we can assume that $\mathcal{A}$ is deterministic. We use $\mathsf{S}_1, \ldots, \mathsf{S}_k$ to denote the collections of stochastic sets chosen by the algorithm in each of its k adaptivity rounds. We further use the random variables $\mathsf{V}_1, \ldots, \mathsf{V}_k$ to denote the vertices on the canonical path of $\mathsf{X}^{(k)}$ (note that V_1 is always root of the edifice $\mathcal{T}$).

Let $d := N^2$ denote the number of children any non-leaf vertex has in $\mathcal{T}$. For any $i \in [k-1]$ we define the following two events:

Event $\mathcal{E}_{\mathsf{small}}(i)$

The collection S_i chosen by $\mathcal{A}$ in round i has size $|\mathsf{S}_i| \leq N/8k$.

The event $\mathcal{E}_{\mathsf{small}}(i)$ is only a function of the realizations of first $i-1$ sets $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$ chosen by $\mathcal{A}$ in the first $i-1$ rounds plus the sets visited in round i and their realizations before reaching the threshold fixed by the algorithm to stop the round.

Event $\mathcal{E}_{\mathsf{hit}}(i)$

The collection S_i chosen by $\mathcal{A}$ in round i contains no set X_u where u is a descendant of $v_{i+1} = \mathsf{V}_{i+1}$, i.e., the $(i+1)$-th vertex in the canonical path of $\mathsf{X}^{(k)}$

The event $\mathcal{E}_{\mathsf{hit}}(i)$ is also only a function of the realizations of the first $i-1$ sets $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}, \mathsf{S}_i$, as well as $\mathsf{V}_1, \ldots, \mathsf{V}_{i+1}$.

The following claim implies that event $\mathcal{E}_{\mathsf{small}}(i)$ is most likely to result in $\mathcal{E}_{\mathsf{hit}}(i)$ as well.

Claim 7.8.6. *For any $i \in [k-1]$,* $\Pr\left(\mathcal{E}_{\mathsf{hit}}(i) \mid \mathcal{E}_{\mathsf{small}}(1), \ldots, \mathcal{E}_{\mathsf{small}}(i), \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)\right) \geq 1 - \frac{1}{2k}$.

Proof. Let $v_1, \ldots, v_i$ be the first i vertices on the canonical path of $\mathsf{X}^{(k)}$. By definition of events $\mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)$, and since v_i is a descendent of all $v_1, \ldots, v_{i-1}$ by definition, we know that no set X_v belong to $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$ for any descendent v of v_i. In particular, X_{v_i} has not been chosen in $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$ and hence its distribution conditioned on $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$ is still the same distribution as before. As such, the $(i+1)$-vertex of the canonical path of $\mathsf{X}^{(k)}$, i.e., v_{i+1} is still chosen uniformly at random over the child-vertices of v_i, even conditioned on the realizations of $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$. On the other hand, conditioned on realizations of $\mathsf{S}_1, \ldots, \mathsf{S}_{i-1}$, the ordering for set S_i chosen by $\mathcal{A}$ is determined deterministically. Let $\widetilde{\mathsf{S}}$ be the set of first $N/8k$ (as in event $\mathcal{E}_{\mathsf{small}}(i)$) items in S_i.

For any $j \in [|\widetilde{\mathsf{S}}|]$, we define an indicator random variable $Y_j \in \{0, 1\}$ which is 1 iff the j-th set chosen in $\widetilde{\mathsf{S}}$ is some X_v for a descendent v of v_{i+1} (notice that this event is based on the set of items chosen in $\widetilde{\mathsf{S}}$ *not their realizations*). Let $u_1, \ldots, u_d$ be the d child-vertices of v_i. We have,

$$\Pr_{v_{i+1}}\left(Y_j = 1 \mid \mathcal{E}_{\mathsf{small}}(1), \ldots, \mathcal{E}_{\mathsf{small}}(i), \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)\right) \leq \frac{1}{d}. \tag{7.8.1}$$

This is simply because only $1/d$ fraction of descendants of v_i are also descendent of v_{i+1} as $\mathcal{T}$ is a d-ary tree. Define $Y = \sum_{j=1}^{|\widetilde{\mathsf{S}}|} Y_j$, i.e., the number of sets chosen from a descendent of v_{i+1}:

$$\begin{aligned}
&\Pr\left(Y \geq 1 \mid \mathcal{E}_{\mathsf{small}}(1), \ldots, \mathcal{E}_{\mathsf{small}}(i), \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)\right) \\
&\qquad \leq \mathbb{E}\left[Y \mid \mathcal{E}_{\mathsf{small}}(1), \ldots, \mathcal{E}_{\mathsf{small}}(i), \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)\right] && \text{(Markov inequality)} \\
&\qquad \underset{\text{Eq (7.8.1)}}{\leq} \frac{|\widetilde{\mathsf{S}}|}{d} \leq \frac{1}{8k}. && \text{(as } d = N^2 \text{ and } |\widetilde{\mathsf{S}}| \leq N/8k \text{ and } N \geq 1\text{)}
\end{aligned}$$

Now notice that under event $\mathcal{E}_{\mathsf{small}}(i)$, in the i-th round, we only pick the sets that are in $\widetilde{\mathsf{S}}$ and hence under this conditioning, the probability that any descendants of v_{i+1} belongs to $\widetilde{\mathsf{S}}_i$ is at most $1/8k$. This concludes the proof. ■

Define the events $\mathcal{E}_{\mathsf{small}}(*) := \mathcal{E}_{\mathsf{small}}(1), \ldots, \mathcal{E}_{\mathsf{small}}(k-1)$ and $\mathcal{E}_{\mathsf{hit}}(*) := \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(k-1)$. We now prove that conditioned on these two events, expected cost of $\mathcal{A}$ is large, in particular S_k needs to be large in expectation.

Lemma 7.8.7. $\mathbb{E}_{S_1,\ldots,S_{k-1}} \mathbb{E}_{\mathsf{S}_k}\left[|\mathsf{S}_k| \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)\right] = \Omega(N/k)$.

Proof. Fix any $S_1, \ldots, S_{k-1}$ conditioned on events $\mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)$; as argued before, these events are only a function $S_1, \ldots, S_{k-1}$. We now bound $|\mathsf{S}_k|$ in expectation.

Recall that v_k is the k-th vertex of the canonical path of $\mathsf{X}^{(k)}$ which is a leaf vertex of $\mathcal{T}$. By event $\mathcal{E}_{\mathsf{hit}}(*)$, we know that X_{v_k} has not been chosen by $\mathcal{A}$ in $S_1, \ldots, S_{k-1}$. As such, conditioned on $(S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*))$, the set X_{v_k} still realizes to some set $U_{v_k} \setminus \{e^\star\}$ for $e^\star \in U_{v_k}$ uniformly at random. In particular, for any element $e \in U_{v_k}$,

$$\Pr_{e^\star}\left(e^\star = e \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)\right) = \frac{1}{|U_{v_k}|}. \tag{7.8.2}$$

Let U_{cov} be the set of elements covered in the first $k-1$ rounds, i.e., by $S_1, \ldots, S_{k-1}$. Let $U'_{v_k} := U_{v_k} \setminus U_{cov}$ be the set of elements in U_{v_k} which are *not* covered in the first $k-1$ rounds.

As $S_1, \ldots, S_{k-1}$ do not contain X_{v_k}, we can apply Lemma 7.8.5 and obtain that

$$\begin{aligned} \left|U'_{v_k}\right| &= |U_{v_k}| - |U_{v_k} \cap U_{cov}| && (7.8.3)\\ &\underset{\text{Lemma 7.8.5}}{\geq} |U_{v_k}| - \sum_{i=1}^{k-1} |S_i| \cdot 2k \geq N - (N/8k) \cdot 4k && (7.8.4)\\ &= N/2, && (7.8.5) \end{aligned}$$

as by event $\mathcal{E}_{\mathsf{small}}(*)$, $|S_i| \leq N/8k$ for all $i \in [k-1]$.

Conditioned on $S_1, \ldots, S_{k-1}$, the ordering chosen for S_k is fixed. Let $\tau := N/16k$ and $\mathsf{X}_1, \ldots, \mathsf{X}_\tau$ be the first τ sets in this ordering. Now consider the element $\{e^\star\} = U_{v_k} \setminus \mathsf{X}_{v_k}$; this element is chosen uniformly at random from U_{v_k} as argued before. We lower bound the probability that the first τ sets in S_k can cover this element $e^\star$. Clearly X_{v_k} cannot cover $e^\star$, hence in the following, without loss of generality, we assume that $\mathsf{X}_1, \ldots, \mathsf{X}_\tau$ do not contain X_{v_k}. This together with Lemma 7.8.5 implies that $|(\mathsf{X}_1 \cup \ldots \mathsf{X}_k) \cap U_{v_k}| \leq \tau \cdot 4k$. We have,

$$\begin{aligned} &\Pr\left(e^\star \in U_{cov} \cup \mathsf{X}_1 \cup \ldots \cup \mathsf{X}_\tau \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)\right)\\ &\underset{\text{Eq (7.8.2)}}{\leq} \frac{|U_{cov}|}{U_{v_k}} + \frac{|(\mathsf{X}_1 \cup \ldots \cup \mathsf{X}_{k-1}) \cap U_{v_k}|}{|U_{v_k}|}\\ &\underset{\text{Eq (7.8.5)}}{\leq} \frac{N}{2N} + \frac{\tau \cdot 4k}{N} = \frac{3}{4}. \end{aligned}$$

(by choice of $\tau = N/16k$ and since $|U_{v_k}| = N$ by Property (II) of edifice in Definition 7.8.2)

This means that with probability at least 1/4, S_k needs to pick more than τ sets to cover the universe U (in particular the element $e^\star$), hence,

$$\mathop{\mathbb{E}}_{\mathsf{S}_k}\left[|\mathsf{S}_k| \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)\right] \geq \tau/4 = \Omega(N/k).$$

Taking an expectation over $S_1, \ldots, S_{k-1}$ conditioned on $\mathcal{E}_{\mathsf{small}}(*), \mathcal{E}_{\mathsf{hit}}(*)$ concludes the proof. ▮

We are now ready to finalize the proof.

Lemma 7.8.8. $\mathbb{E}_{X\sim\mathsf{X}^{(k)}}\left[\mathcal{A}(X)\right] = \Omega(N/k^2)$.

Proof. We can write the expected cost of $\mathcal{A}$ as:

$$\begin{aligned}
\mathop{\mathbb{E}}_{X\sim\mathsf{X}^{(k)}}\left[\mathcal{A}(X)\right] &= \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1\Big] \\
&= \Pr\left(\mathcal{E}_{\mathsf{small}}(1)\right)\cdot \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\qquad + \left(1-\Pr\left(\mathcal{E}_{\mathsf{small}}(1)\right)\right)\cdot \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\overline{\mathcal{E}_{\mathsf{small}}(1)}\Big] \\
&\geq \Pr\left(\mathcal{E}_{\mathsf{small}}(1)\right)\cdot \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\qquad + \left(1-\Pr\left(\mathcal{E}_{\mathsf{small}}(1)\right)\right)\cdot N/8k.
\end{aligned}$$

The inequality is by definition of $\overline{\mathcal{E}_{\mathsf{small}}(1)}$ as this means that $|S_1| \geq N/8k$. As such, if $\Pr\left(\mathcal{E}_{\mathsf{small}}(*)\right) \leq (1-1/2k)$, we are already done as in this case the second term in RHS above is at least $(N/8k)\cdot(1/2k) = \Omega(N/k^2)$. Otherwise,

$$\begin{aligned}
\mathop{\mathbb{E}}_{X\sim\mathsf{X}^{(k)}}\left[\mathcal{A}(X)\right] &\geq (1-1/2k)\cdot \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\geq (1-1/2k)\cdot\Pr\left(\mathcal{E}_{\mathsf{hit}}(1)\mid\mathcal{E}_{\mathsf{small}}(1)\right)\mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\underset{\text{Claim 7.8.6}}{\geq} (1-1/2k)^2\cdot \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big].
\end{aligned}$$

We now continue this calculation for the RHS using the sets S_2 in second round:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_1,\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\quad= \mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{S_2}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_2,S_1,\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\quad= \Pr\left(\mathcal{E}_{\mathsf{small}}(2)\mid\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\right)\mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{S_2}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_2,S_1,\mathcal{E}_{\mathsf{small}}(2),\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big] \\
&\qquad\quad + \Pr\left(\overline{\mathcal{E}_{\mathsf{small}}(2)}\mid\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\right)\cdot\mathop{\mathbb{E}}_{S_1}\mathop{\mathbb{E}}_{S_2}\mathop{\mathbb{E}}_{X}\Big[\mathcal{A}(X)\mid S_2,S_1,\overline{\mathcal{E}_{\mathsf{small}}(2)},\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\Big]
\end{aligned}$$

Again, if $\Pr\left(\mathcal{E}_{\mathsf{small}}(2)\mid\mathcal{E}_{\mathsf{hit}}(1),\mathcal{E}_{\mathsf{small}}(1)\right) \leq (1-1/2k)$, we are already done as in this case the second term in RHS above is at least $\Omega(N/k^2)$. Combining this with previous equation, we obtain that expected cost of $\mathcal{A}$ is at least $(1-1/2k)^3 \cdot \Omega(N/k^2) = \Omega(N/k^2)$. Hence, we can assume that $\Pr\left(\mathcal{E}_{\mathsf{small}}(2) \mid \mathcal{E}_{\mathsf{hit}}(1), \mathcal{E}_{\mathsf{small}}(1)\right) \geq (1-1/2k)$. Using this, and the previous argument we did for the first round, and by Claim 7.8.6, we obtain that:

$$\mathop{\mathbb{E}}_{X \sim \mathsf{X}^{(k)}}[\mathcal{A}(X)] \geq \left(1 - \frac{1}{2k}\right)^4 \cdot \mathop{\mathbb{E}}_{S_1} \mathop{\mathbb{E}}_{S_2} \mathop{\mathbb{E}}_{X} \Big[\mathcal{A}(X) \mid S_2, S_1, \mathcal{E}_{\mathsf{hit}}(2), \mathcal{E}_{\mathsf{small}}(2), \mathcal{E}_{\mathsf{hit}}(1), \mathcal{E}_{\mathsf{small}}(1)\Big].$$

We can thus continue this argument until processing the last round, and either we already have $\mathbb{E}_{X \sim \mathsf{X}^{(k)}} = \Omega(N/k^2)$ as for some $i \in [k-1]$, $\Pr\left(\mathcal{E}_{\mathsf{small}}(i) \mid 1, \ldots, \mathcal{E}_{\mathsf{small}}(i-1), \mathcal{E}_{\mathsf{hit}}(1), \ldots, \mathcal{E}_{\mathsf{hit}}(i-1)\right)$ is greater than or equal to $(1-1/2k)$, or:

$$\begin{aligned}
\mathop{\mathbb{E}}_{X \sim \mathsf{X}^{(k)}}[\mathcal{A}(X)] &\geq \left(1 - \frac{1}{2k}\right)^{2k-2} \cdot \mathop{\mathbb{E}}_{S_1,\ldots,S_{k-1}} \mathop{\mathbb{E}}_{X} \Big[\mathcal{A}(X) \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{hit}}(*), \mathcal{E}_{\mathsf{small}}(*)\Big] \\
&\geq \Omega(1) \cdot \mathop{\mathbb{E}}_{S_1,\ldots,S_{k-1}} \mathop{\mathbb{E}}_{\mathsf{S}_k} \left[|\mathsf{S}_k| \mid S_1, \ldots, S_{k-1}, \mathcal{E}_{\mathsf{hit}}(*), \mathcal{E}_{\mathsf{small}}(*)\right] \\
&\mathop{\geq}_{\text{Lemma 7.8.7}} \Omega(N/k).
\end{aligned}$$

This concludes the proof. ■

Theorem 7.8.1 now follows from Lemma 7.8.8 and Claim 7.8.4, by setting $r = k$ and noticing that $N = n^{1/k}$ in this construction.

Chapter 8

Conclusion

I believe that in order to develop machine learning into a rich scientific discipline we need to create bridges for two-way exchange of ideas between machine learning and other disciplines that allow us to develop principled solutions to common problems. My research has contributed towards the creation of these two-way bridges between machine learning and information elicitation/mechanism design, choice/preference elicitation, and theoretical computer science. In the future, I hope to explore more problems at these interfaces and further contribute towards exchange of ideas between these fields.

APPENDIX

A.1 Appendix to Chapter 4

A.1.1 Generalization of the ASR algorithm with Regularization

In this section, we shall present a generalized version of the ASR algorithm that relaxes the assumption that each set S_a is of the same fixed cardinality m, and each set S_a is compared the same number of times L. The intuition behind this generalization is that each comparison carries an equal amount of information, and thus, we should give a higher preference to the empirical estimates $\widehat{p}_{i|S_a}$ corresponding to sets with more comparisons. Furthermore, comparisons on smaller sets are more reliable than comparisons on larger sets. In general, sets with larger cardinality should have proportionately more comparisons. Lastly, in practice, we often encounter comparison data for which the random walk $\widehat{\mathbf{P}}$ on the comparison graph G_c is not strongly connected. We can resolve this issue through regularization. With these in mind, we update our algorithm as discussed below:

Given general comparison data $\mathbf{Y'} = \{(S_a, \mathbf{y}_a)_{a=1}^d\}$, where $S_a \subseteq [n]$ is of cardinality $|S_a|$, and $\mathbf{y}_a = (y_a^1, \ldots, y_a^{L_a})$, we define d'_i for each $i \in [n]$ as

$$d'_i := \sum_{a \in [d] : i \in S_a} \left(\frac{L_a}{|S_a|} + \lambda \right)$$

where λ is a regularization parameter. Intuitively, one can think of the regularization as adding $\lambda|S_a|$ pseudo-comparisons to each set S_a, with each item in the set winning an equal λ times. Furthermore, we define $n_{i|S_a}$ to be the number of times item $i \in S_a$ won in a $|S_a|$-way comparison amongst items in S_a, i.e. for all $a \in [d]$, for all $i \in S_a$,

$$n_{i|S_a} := \sum_{l=1}^{L_a} \mathbf{1}[y_a^l = i] \tag{A.1.1}$$

Algorithm 10 Generalized-ASR

```
Input Markov chain $\widehat{\mathbf{P}}'$ (according to Eq. (A.1.2))
Initialize $\widehat{\boldsymbol{\pi}} = (\frac{1}{n}, \cdots, \frac{1}{n})^\top \in \Delta_n$
while estimates do not converge do
    $\widehat{\boldsymbol{\pi}}' \leftarrow \widehat{\mathbf{P}}'^\top \widehat{\boldsymbol{\pi}}'$
end while
Output $\widehat{\mathbf{w}}' = \frac{\mathcal{D}'^{-1}\widehat{\boldsymbol{\pi}}'}{\|\mathcal{D}'^{-1}\widehat{\boldsymbol{\pi}}'\|_1}$
```

Using the above notation, we set up a Markov chain $\widehat{\mathbf{P}}' \in \mathbb{R}_+^{n\times n}$ such that entry (i, j) is

$$\widehat{P}'_{ij} := \frac{1}{d'_i} \sum_{a\in[d]:i,j\in S_a} \left(\frac{n_{j|S_a} + \lambda}{|S_a|} \right) \tag{A.1.2}$$

One can verify that this non-negative matrix is indeed row stochastic, hence corresponds to the transition matrix of a Markov chain. One can also verify that this construction reduces to a regularized version of $\widehat{\mathbf{P}}$ (Eq. (4.3.2)) when all sets are of an equal size and are compared an equal number of times, and is identical to $\widehat{\mathbf{P}}$ when $\lambda = 0$. Lastly, we define the matrix $\mathcal{D}'$ as a diagonal matrix, with diagonal entry $D'_{ii} := d'_i$, $\forall i \in [n]$. Similar to ASR, we compute the stationary distribution of $\widehat{\mathbf{P}}'$, and output a (normalized) $\mathcal{D}'^{-1}$ transform of this stationary distribution.

A.1.2 Proof of Corollary 4.5.7

Corollary 4.5.7 follows from the following lemma which compares the spectral gap of the matrix $\mathbf{P}$ with the spectral gap of the graph Laplacian.

Lemma A.1.1. *Let $\mathbf{L} := \mathbf{C}^{-1}\mathbf{A}$ be the Laplacian of the undirected graph $G_c([n], E)$. Then the spectral gap $\mu(\mathbf{P}) = 1-\lambda_2(\mathbf{P})$ of the reversible Markov chain $\mathbf{P}$ (Eq. (4.3.2)) corresponding to the* ASR *algorithm is related to the spectral gap $\xi = 1 - \lambda_2(\mathbf{L})$ of the Laplacian as*

$$\mu(\mathbf{P}) \geq \frac{\xi}{mb^2}$$

Proof. To prove this inequality, we shall leverage the comparison Lemma 4.4.4 of Diaconis and Saloff-Coste (1993), with $\mathbf{Q}, \boldsymbol{\nu} = \mathbf{L}, \boldsymbol{\nu}$. From the definition of the Laplacian, it is clear that for all i, $\nu_i \mathbf{L}_{ij} = 1/2|E|$. Furthermore, $\nu_i = c_i/2|E| \geq d_i/2|E|$, where c_i is the number of unique items i was compared with, which is trivially at least the number of unique multiway comparisons of which i was a part. Thus,

$$\begin{aligned}
\beta &:= \max_{i\in[n]} \frac{\pi_i}{\nu_i} = \max_{i\in[n]} \frac{w_i d_i/\|\mathcal{D}\mathbf{w}\|_1}{c_i/2|E|} \\
&\leq \frac{2|E|w_{\max}}{\|\mathcal{D}\mathbf{w}\|_1} \\
\alpha &:= \min_{(i,j)\in E} \frac{\pi_i P_{ij}}{\nu_i L_{ij}} \\
&= \min_{(i,j)\in E} \frac{\frac{w_i d_i}{\|\mathcal{D}\mathbf{w}\|_1} \frac{1}{d_i} \sum_{a:(i,j)\in S_a} \frac{w_j}{\sum_{k\in S_a} w_k}}{1/2|E|} \\
&\geq \frac{2|E|w_{\min}^2}{m w_{\max}\|\mathcal{D}\mathbf{w}\|_1}
\end{aligned}$$

Thus, $\alpha/\beta \geq 1/mb^2$, which proves our claim. □

A.1.3 Proof of Corollary 4.5.8

In order to prove this corollary we first give the following claim.

Claim A.1.2. *Given items $[n]$, and comparison graph $G_c = ([n], E)$ induced by comparison data $\mathbf{Y} = \{S_a, \mathbf{y}_a\}_{a=1}^d$, let the vector of true MNL parameters be $\mathbf{w} = (w_1, \ldots, w_n)$. Furthermore, let d_i represent the number of unique comparisons of which item $i \in [n]$ was a part. Then we have*

$$d_{\text{avg}} = \sum_{i\in[n]} w_i d_i \leq \frac{2w_{\max}|E|}{w_{\min} n} \,,$$

where $w_{\max} = \max_{i\in[n]} w_i$, and $w_{\min} = \min_{j\in[n]} w_j$.

Proof. Clearly,

$$w_{\min} \sum_{i\in[n]} w_i d_i \leq \frac{1}{n} \sum_{i\in[n]} w_i d_i \leq \frac{w_{\max}}{n} \sum_{i\in[n]} d_i,$$

The statement of the lemma follows by realizing that $\sum_{i\in[n]} d_i \leq \sum_{i\in[n]} c_i \leq 2|E|$. □

Proof. (of Corollary 4.5.8) Substituting the above bound on d_{avg} in the sample complexity bounds of Corollary 4.5.7, we get the following guarantee on the total variation error between the estimates $\widehat{\mathbf{w}}$ and the true weight vector $\mathbf{w}$

$$\|\mathbf{w} - \widehat{\mathbf{w}}\|_{\mathrm{TV}} \le \frac{C\, m\, b^3\, \kappa\, |E|}{n\, \xi\, d_{\min}} \sqrt{\frac{\max\{m, \log(n)\}}{L}}\,,$$

where $b = \frac{w_{\max}}{w_{\min}}$. Furthermore, this guarantee holds with probability $\ge 1 - 3n^{-(C^2-50)/25}$. From this, we can conclude that if

$$L \ge \max\{m, \log(n)\} \left(\frac{10\, m\, b^3\, \kappa\, |E|}{n\, \xi\, d_{\min}} \right)^2 ,$$

then it is sufficient to guarantee that $\|\mathbf{w} - \widehat{\mathbf{w}}\|_{\mathrm{TV}} = o(1)$ with probability $\ge 1 - 3n^{-2}$. Trivially bounding $\kappa = O(\log n)$, and from the assumptions $b = O(1)$ and $|E| = O(n\,\mathrm{poly}(\log n))$, we can conclude

$$L = O(\xi^{-2} m^3 \mathrm{poly}(\log n))$$

where the additional m factor comes from trivially bounding $\max\{m, \log n\} \le m \log n$. This gives us a sample complexity bound of

$$|E| \times L = O(\xi^{-2}\, m^3\, n\, \mathrm{poly}(\log n))$$

for our algorithm, which proves the corollary. □

A.1.4 Additional Experimental Results

In this section we will describe additional experimental results comparing our algorithm and the RC/LSR algorithms on various synthetic and real world datasets. Since we require additional regularization when the random walk induced by comparison data is reducible, we will first describe the regularized version of the RC and LSR algorithms (regularized version of our algorithm is given in Appendix A.1.1).

A.1.5 RC and LSR algorithms with regularization

In this section, for the sake of completeness, we state the regularized version of the RC (Negahban et al., 2017) and LSR (Maystre and Grossglauser, 2015) algorithms.[1] These algorithms are based on computing the stationary distribution of a Markov chain. In the case of pairwise comparisons, for a regularization parameter $\lambda > 0$, the Markov chain $\widehat{\mathbf{P}}^{\prime\text{RC}} := [\widehat{P}_{ij}^{\prime\text{RC}}]$, where, $\forall i, j \in [n]$,

$$\widehat{P}_{ij}^{\prime\text{RC}} := \begin{cases} \frac{1}{d_{\max}} \left(\frac{n_{j|\{i,j\}} + \lambda}{n_{j|\{i,j\}} + n_{i|\{i,j\}} + 2\lambda} \right), & \text{if } i \neq j \\ 1 - \frac{1}{d_{\max}} \sum_{j' \neq i} \widehat{P}_{ij}^{\prime\text{RC}}, & \text{if } i = j \end{cases}$$

and $n_{j|\{i,j\}}$ is defined according to Eq. (A.1.1). In the case of multi-way comparisons, the Markov chain $\widehat{\mathbf{P}}^{\prime\text{LSR}} := [\widehat{P}_{ij}^{\prime\text{LSR}}]$, where, $\forall i, j \in [n]$,

$$\widehat{P}_{ij}^{\prime\text{LSR}} := \begin{cases} \epsilon \sum_{a \in [d] : i, j \in S_a} \left(\frac{n_{j|S_a} + \lambda}{|S_a|} \right), & \text{if } i \neq j \\ 1 - \epsilon \sum_{j' \neq i} \widehat{P}_{ij}^{\prime\text{LSR}}, & \text{if } i = j \end{cases}$$

where ϵ is a quantity small enough to make the diagonal entries of $\widehat{\mathbf{P}}^{\prime\text{LSR}}$ non negative, and $n_{j|S_a}$ is again defined according to Eq. (A.1.1).

A.1.6 Synthetic Datasets

In this section, we give additional experimental results for various other values of parameters m and n. The plots are given in the figures below. The general trends observed from these experiments are exactly as predicted by our theoretical analysis. In particular, we note that even in the case of a star graph topology, the convergence rate of ASR remains essentially the same with increasing n, while the performance of RC and LSR degrades smoothly. This really conveys the low dependence on the ratio $d_{\max}/d_{\min}$.

[1] See Section 3.3 in Negahban et al. (2017) for more details.

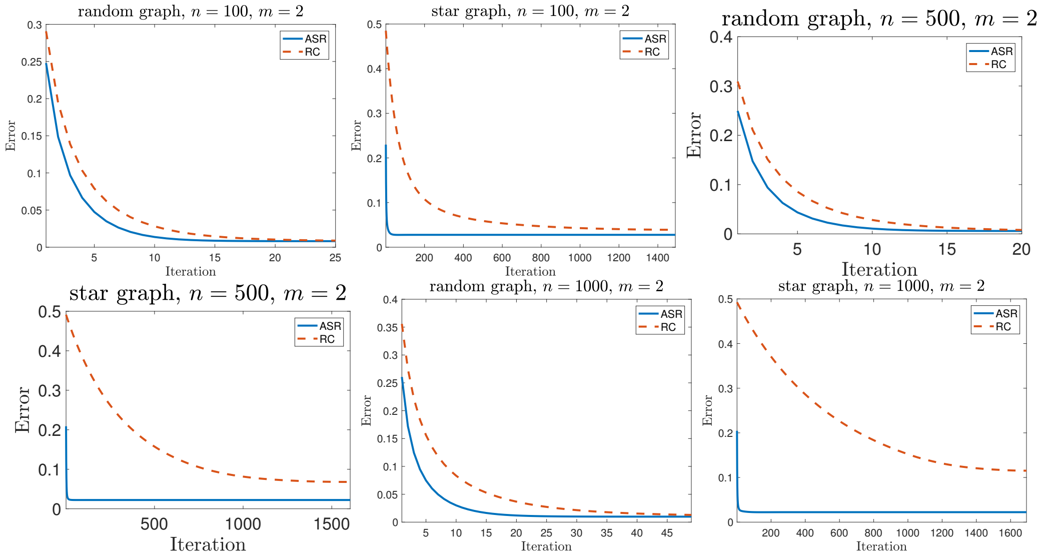


Figure 15: Results on synthetic data: L_1 error vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for $m = 2$) on data generated from the MNL/BTL model with the random and star graph topologies.

A.1.7 Real Datasets

In this section, we provide additional experimental results for more datasets, and additional values of the regularization parameter λ. We conducted experiments on the YouTube dataset (Shetty, 2012), various GIF datasets (Rich et al.), and the SFwork and SFshop (Koppelman and Bhat, 2006) datasets. Below we briefly describe each of these datasets (additional statistics are given in Table 6).

1. **YouTube Comedy Slam Preference Data**. This dataset is due to a video discovery experiment on YouTube in which users were shown a pair of videos and were asked to vote for the video they found funnier out of the two.[2]

2. **GIFGIF datasets**. These datasets are due to a experiment that tries to understand the emotional content present in animated GIFs. In this experiment users are shown a pair of GIFs and asked to vote for the GIF that most accurately represents a particular

[2] See https://archive.ics.uci.edu/ml/datasets/YouTube+Comedy+Slam+Preference+Data for more details.

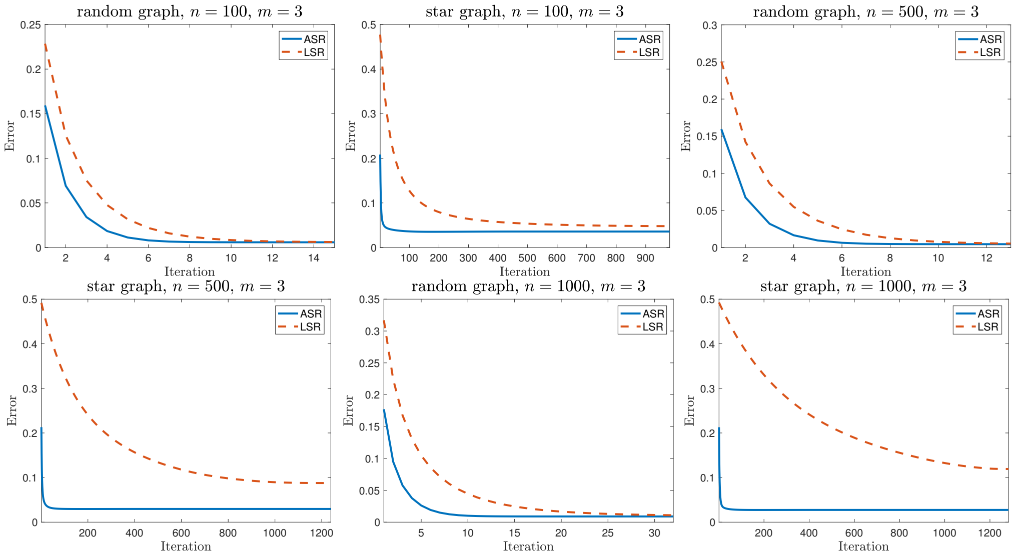


Figure 16: Results on synthetic data: L_1 error vs. number of iterations for our algorithm, ASR, compared with the LSR algorithm (for $m = 3$) on data generated from the MNL/BTL model with the random and star graph topologies.

emotion. These votes are collected for several different emotions.[3]

3. **SF datasets**. These datasets are from a survey of transportation preferences around the San Francisco Bay Area in which citizens were asked to vote on their preferred commute option amongst different options.[4]

As expected, the peak log likelihood decreases with increasing λ, as this regularization parameter essentially dampens the information imparted by the comparison data. We also plot degree distributions of these real world datasets in order to explore the behavior of the ratio $d_{\max}/d_{\min}$ in practice. In particular, we observe that this quantity does not really behave like a constant, and is very large in most cases. This is particularly evident in the Youtube dataset, where the degree distribution closely follows the power law relationship with n.

[3] See `http://gif.gf` for more details.

[4] These datasets are available at `https://github.com/sragain/pcmc-nips`.

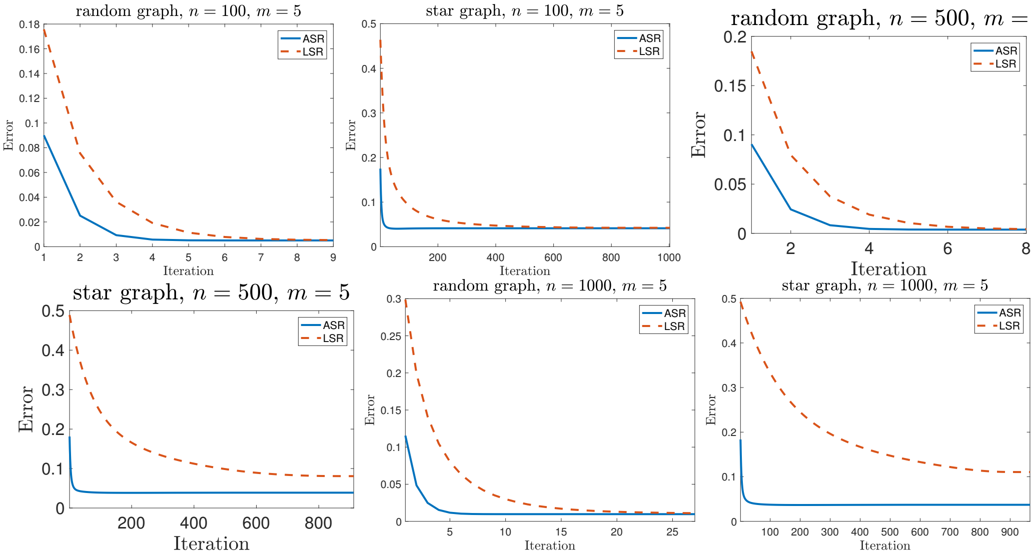


Figure 17: Results on synthetic data: L_1 error vs. number of iterations for our algorithm, ASR, compared with the LSR algorithm (for $m = 5$) on data generated from the MNL/BTL model with the random and star graph topologies.

Table 6: Statistics for real world datasets

Dataset	n	m	d	**total choices**
Youtube	21207	2	394007	1138562
GIF-amusement	6118	2	75649	77609
GIF-anger	6119	2	64830	66505
GIF-contentment	6118	2	70230	72175
GIF-excitement	6119	2	80493	82564
GIF-happiness	6119	2	104801	107816
GIF-pleasure	6119	2	86499	88959
GIF-relief	6112	2	38770	39853
GIF-sadness	6118	2	63577	65263
GIF-satisfaction	6118	2	78401	80474
GIF-shame	6116	2	46249	47550
GIF-surprise	6118	2	63850	65591
SFWork	6	3-6	12	5029
SFShop	8	4-8	10	3157

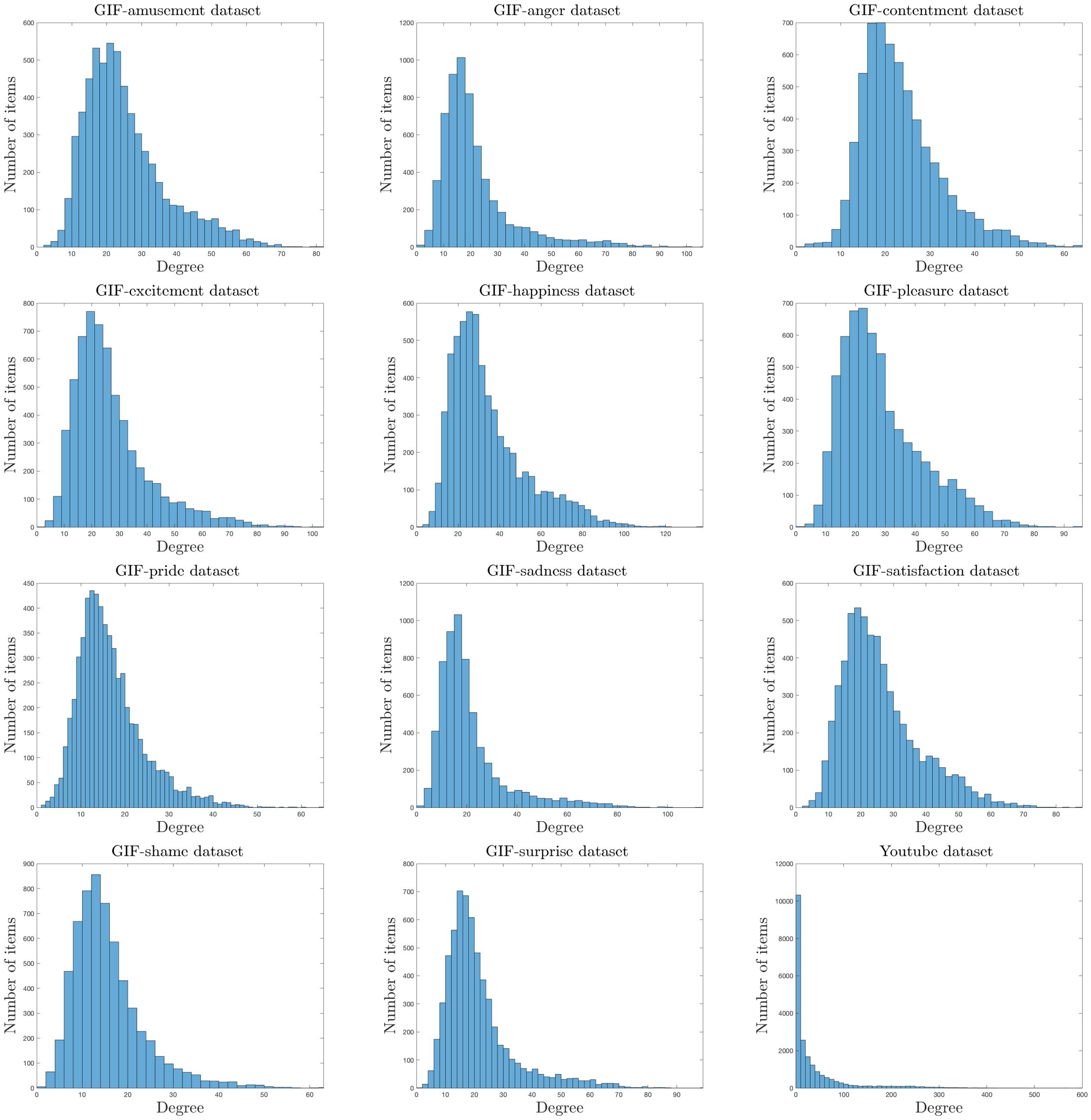


Figure 18: Degree distributions of various real world datasets.

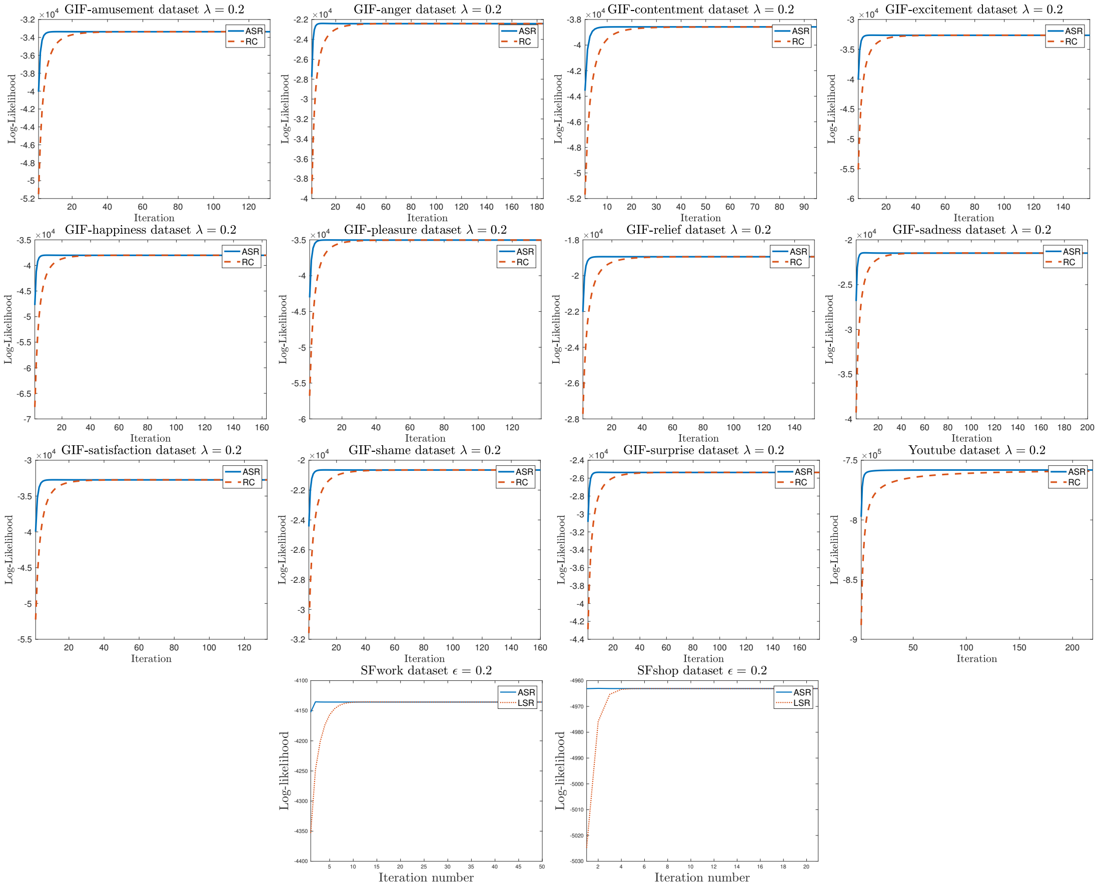
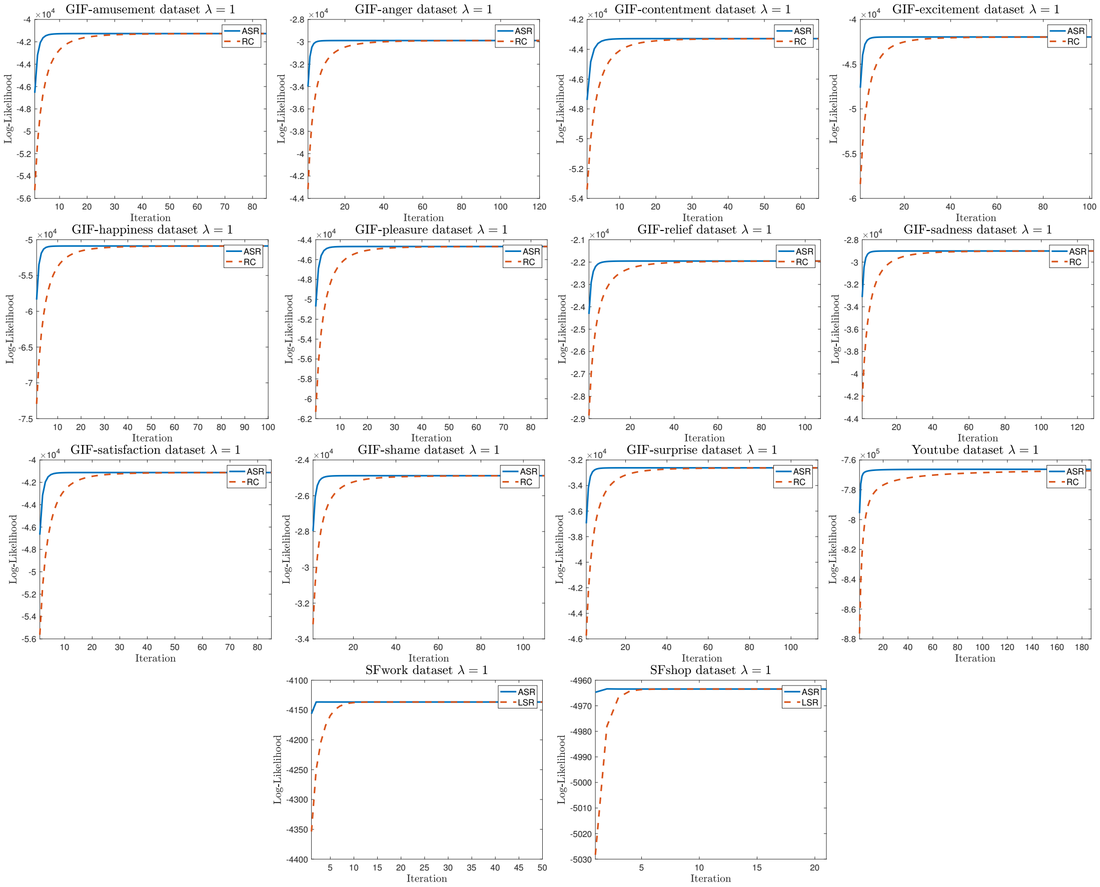


Figure 19: Results on real data: Log-likelihood vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for pairwise comparison data) and the LSR algorithm (for multi-way comparison data), all with regularization parameter set to 0.2.


GIF-amusement dataset $\lambda = 1$

GIF-anger dataset $\lambda = 1$

GIF-contentment dataset $\lambda = 1$

GIF-excitement dataset $\lambda = 1$

GIF-happiness dataset $\lambda = 1$

GIF-pleasure dataset $\lambda = 1$

GIF-relief dataset $\lambda = 1$

GIF-sadness dataset $\lambda = 1$

GIF-satisfaction dataset $\lambda = 1$

GIF-shame dataset $\lambda = 1$

GIF-surprise dataset $\lambda = 1$

Youtube dataset $\lambda = 1$

SFwork dataset $\lambda = 1$

SFshop dataset $\lambda = 1$

Figure 20: Results on real data: Log-likelihood vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for pairwise comparison data) and the LSR algorithm (for multi-way comparison data), all with regularization parameter set to 1.

A.2 Appendix to Chapter 5

A.2.1 Estimation of Choice Models from Real-World Datasets

We estimate choice probabilities from several real-world preference datasets, which contain multiple partial preference orders over items. The choice probability $P_{i|S}$ of an item i over S, was taken to be the fraction of times in these partial order item i was the top ranked items in S. More formally, let there be m partial orders, $\mathcal{P}_1, \cdots, \mathcal{P}_m$, over n items. For any subset $S \subseteq [n]$, and $i \in [n]$, let $N_{i|S}$ be defined as:

$$N_{i|S} := \sum_{j \in [m]} \mathbb{1}[\forall i' \in S \setminus \{i\} : i \succ_{\mathcal{P}_j} i'] \,.$$

The choice probability $P_{i|S}$ is then estimated as:

$$P_{i|S} := \frac{N_{i|S}}{\sum_{i' \in S} N_{i'|S}} \,.$$

A.2.2 Runtime and Space Complexity of WBA-A and WBA-L

The space complexity of our algorithms is $O(n^2)$ as they only store the pairwise statistics extracted from multiway choices. Each trial in our algorithms runs in time polynomial in n. The most non-trivial step is computing $\mathcal{J}_i(t, C)$ for each arm. This step requires polynomial time because we can compute the quantity $\text{argmax}_{S \subseteq [n]} I_i(t, S) - |S| \cdot \log(nC)$ and check if it is greater than $\log(t)$. We compute $\text{argmax}_{S \subseteq [n]} I_i(t, S) - |S| \cdot \log(nC)$ by first sorting arms j in the order of values $\mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2})$. We then start with $S \leftarrow \emptyset$ and add one arm at a time from this sorted ordering to S. We stop adding arms to the set S once the value $\mathbb{1}[\widehat{P}_{ij}(t) \leq \frac{1}{2}] \cdot N_{ij}(t) \cdot d(\widehat{P}_{ij}(t), \frac{1}{2})$ of the current arm j is less than $\log(nC)$. It is easy to see that computing $I_i(t, S) - |S| \cdot \log(nC)$ for this set S gives the value of $\text{argmax}_{S \subseteq [n]} I_i(t, S) - |S| \cdot \log(nC)$.

A.2.3 Technical Lemmas

Theorem A.2.1. *[Bernstein Inequality for Martingales; Cesa-Bianchi and Lugosi (2006)] Let* $X_1, ..., X_m$ *be a bounded martingale difference sequence with respect to the filtration* $\mathcal{F} = (\mathcal{F}_i)_{1 \leq i \leq m}$ *and with* $|X_i| \leq K$*. Let* $Z_i = \sum_{j=1}^{i} X_j$ *be the associated martingale sequence. Let the sum of the conditional variances be* $\Sigma_m^2 = \sum_{i=1}^{m} \mathbf{E}[X_i^2 | \mathcal{F}_{i-1}]$*. Then for all constants* $\lambda, \nu > 0$*,*

$$\Pr\left(\max_{i \in [m]} |Z_i| > \sqrt{2\nu t} + 2Kt/3, \Sigma_m^2 \leq \nu\right) \leq 2e^{-t}\,.$$

BIBLIOGRAPHY

J. D. Abernethy and R. M. Frongillo. A characterization of scoring rules for linear properties. In *Proceedings of the 25th Annual Conference on Learning Theory*, 2012.

A. Agarwal and S. Agarwal. On consistent surrogate risk minimization and property elicitation. In *Conference on Learning Theory*, pages 4–22, 2015.

A. Agarwal, S. Agarwal, S. Assadi, and S. Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 39–75, 2017a.

A. Agarwal, D. Mandal, D. C. Parkes, and N. Shah. Peer prediction with heterogeneous users. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, page 81–98, New York, NY, USA, 2017b. Association for Computing Machinery. ISBN 9781450345279. doi: 10.1145/3033274.3085127. URL https://doi.org/10.1145/3033274.3085127.

A. Agarwal, P. Patil, and S. Agarwal. Accelerated spectral ranking. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 70–79, 2018.

A. Agarwal, S. Assadi, and S. Khanna. Stochastic submodular cover with limited adaptivity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 323–342, 2019a.

A. Agarwal, N. Johnson, and S. Agarwal. Choice bandits. In *preparation*, 2019b.

A. Agarwal, N. Johnson, and S. Agarwal. Choice bandits. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

S. Agrawal, V. Avadhanula, V. Goyal, and A. Zeevi. A Near-Optimal Exploration-Exploitation Approach for Assortment Selection. In *Proceedings of the 17th ACM Conference on Economics and Computation*, 2016.

S. Agrawal, V. Avadhanula, V. Goyal, and A. Zeevi. Thompson sampling for the mnl-bandit. In *COLT*, 2017.

N. Ailon, Z. Karnin, and T. Joachims. Reducing Dueling Bandits to Cardinal Bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

M. Ajtai, J. Komlos, W. L. Steiger, and E. Szemerédi. Deterministic selection in $o(\log \log n)$ parallel time. In *STOC*, 1986.

H. Allcott and M. Gentzkow. Social Media and Fake News in the 2016 Election. Technical report, National Bureau of Economic Research, 2017.

N. Alon and Y. Azar. Sorting, approximate sorting, and searching in rounds. *SIAM J. Discrete Math.*, 1(3):269–280, 1988.

A. Anagnostopoulos, L. Becchetti, I. Bordino, S. Leonardi, I. Mele, and P. Sankowski. Stochastic query covering for fast approximate document retrieval. *ACM Trans. Inf. Syst.*, 33(3):11:1–11:35, Feb. 2015. ISSN 1046-8188. doi: 10.1145/2699671. URL http://doi.acm.org/10.1145/2699671.

A. Anandkumar, R. Ge, D. J. Hsu, S. M. Kakade, and M. Telgarsky. Tensor Decompositions for Learning Latent Variable Models. *Journal of Machine Learning Research*, 15(1): 2773–2832, 2014.

A. Asadpour and H. Nazerzadeh. Maximizing stochastic monotone submodular functions. *Management Science*, 62(8):2374–2391, 2016.

A. Asadpour, H. Nazerzadeh, and A. Saberi. Stochastic submodular maximization. In *Internet and Network Economics, 4th International Workshop, WINE 2008, Shanghai, China, December 17-20, 2008. Proceedings*, pages 477–489, 2008.

S. Assadi and S. Khanna. Tight bounds on the round complexity of the distributed maximum coverage problem. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2412–2431, 2018.

S. Assadi, S. Khanna, and Y. Li. The stochastic matching problem with (very) few queries. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 43–60, 2016.

S. Assadi, S. Khanna, and Y. Li. The stochastic matching problem: Beating half with a non-adaptive algorithm. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 99–116, 2017.

S. Athey and G. W. Imbens. Machine learning methods that economists should know about. *Annual Review of Economics*, 11:685–725, 2019.

J.-Y. Audibert and S. Bubeck. Best Arm Identification in Multi-Armed Bandits. In *COLT*, 2010.

P. Awasthi, A. Blum, O. Sheffet, and A. Vijayaraghavan. Learning mixtures of ranking models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2609–2617, 2014.

Y. Azar and I. Gamzu. Ranking with submodular valuations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1070–1079, 2011.

Y. Azar, I. Gamzu, and X. Yin. Multiple intents re-ranking. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 669–678, 2009.

E. Balkanski and Y. Singer. The adaptive complexity of maximizing a submodular function. In *STOC 2018 (To Appear)*.

E. Balkanski and Y. Singer. A lower bound for parallel submodular minimization. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 130–139. ACM, 2020.

E. Balkanski, A. Rubinstein, and Y. Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. *CoRR*, abs/1804.06355, 2018.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

P. L. Bartlett, M. Jordan, and J. McAuliffe. Convexity, classification and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

V. Bengs and E. Hüllermeier. Preselection bandits under the plackett-luce model. *CoRR*, abs/1907.06123, 2019. URL http://arxiv.org/abs/1907.06123.

A. Blum, J. P. Dickerson, N. Haghtalab, A. D. Procaccia, T. Sandholm, and A. Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 325–342, 2015.

B. Bollobás and G. Brightwell. Parallel selection with high probability. *SIAM Journal on Discrete Mathematics*, 3(1):21–31, 1990.

B. Bollobás and A. Thomason. Parallel sorting. *Discrete Applied Mathematics*, 6(1):1–11, 1983.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952a.

R. A. Bradley and M. E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3-4):324–345, 1952b.

M. Braverman, J. Mao, and S. M. Weinberg. Parallel Algorithms for Select and Partition with Noisy Comparisons. In *STOC*, 2016a.

M. Braverman, J. Mao, and S. M. Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 851–862, 2016b.

M. Braverman, J. Mao, and Y. Peres. Sorted top-k in rounds. In A. Beygelzimer and D. Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 342–382. PMLR, 2019. URL http://proceedings.mlr.press/v99/braverman19a.html.

A. Breuer, E. Balkanski, and Y. Singer. The FAST algorithm for submodular maximization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1134–1143. PMLR, 2020.

B. Brost, Y. Seldin, I. J. Cox, and C. Lioma. Multi-Dueling Bandits and Their Application to Online Ranker Evaluation. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 2016.

S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *ICML*, 2013.

D. Buffoni, C. Calauzènes, P. Gallinari, and N. Usunier. Learning scoring functions with order-preserving losses and standardized supervision. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. 2005.

R. Busa-Fekete, B. Szorenyi, P. Weng, W. Cheng, and E. Hullermeier. Top-k Selection based on Adaptive Sampling of Noisy Preferences. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Y. Cai, C. Daskalakis, and C. Papadimitriou. Optimum statistical estimation with strategic data sources. In *Proceedings of The 28th Conference on Learning Theory*, pages 280–296, 2015.

C. Calauzènes, N. Usunier, and P. Gallinari. On the (non-)existence of convex, calibrated surrogate losses for ranking. In *Advances in Neural Information Processing Systems*, 2012.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.* Cambridge university press, 2006.

A. Chakrabarti and A. Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1365–1373, 2016.

K. Chandrasekaran and R. Karp. Finding a most biased coin with fewest flips. In *Journal of Machine Learning Research*, volume 35, pages 394–407, 2014.

M. Charikar, C. Chekuri, and M. Pál. Sampling bounds for stochastic optimization. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 257–269. Springer, 2005.

B. Chen and P. I. Frazier. Dueling Bandits with Weak Regret. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

L. Chen and J. Li. On the Optimal Sample Complexity for Best Arm Identification. *arXiv preprint arXiv:1511.03774*, 2015. URL http://arxiv.org/abs/1511.03774.

L. Chen, J. Li, and M. Qiao. Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection. *arXiv preprint arXiv:1702.03605*, 2017a. URL https://arxiv.org/abs/1702.03605.

W. Chen, Y. Wang, and Y. Yuan. Combinatorial Multi-Armed Bandit: General Framework, Results and Applications. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

X. Chen and Y. Wang. A Note on Tight Lower Bound for MNL-Bandit Assortment Selection Models. Technical report, arXiv:1709.06109v2, 2017.

X. Chen, Y. Li, and J. Mao. A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *SODA*, 2017b.

X. Chen, Y. Li, and J. Mao. A Nearly Instance Optimal Algorithm for Top-k Ranking under the Multinomial Logit Model. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

Y. Chen and C. Suh. Spectral MLE : Top-K Rank Aggregation from Pairwise Comparisons. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

F. Chierichetti, R. Kumar, and A. Tomkins. Learning a mixture of two multinomial logits. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 960–968. PMLR, 2018.

G. Cho and C. Meyer. Comparison of perturbation bounds for the stationary distribution of a Markov chain. *Linear Algebra and its Applications*, 335(1-3):137–150, 2001.

V. Cohen-Addad, F. Mallmann-Trenn, and C. Mathieu. Instance-optimality in the noisy value-and comparison-model. In S. Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2124–2143. SIAM, 2020.

V. Cohen-Addad, S. Lattanzi, S. Mitrovic, A. Norouzi-Fard, N. Parotsidis, and J. Tarnawski. Correlation clustering in constant many parallel rounds. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2069–2078. PMLR, 2021.

R. Cole. Parallel merge sort. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 511–516, 1986.

R. Cole. Parallel merge sort. *SIAM J. Comput.*, 17(4):770–785, 1988.

R. Combes, M. S. Talebi, A. Proutiere, and M. Lelarge. Combinatorial Bandits Revisited. In *Advances in Neural Information Processing Systems 28*, 2015.

G. Cormode, H. J. Karloff, and A. Wirth. Set cover algorithms for very large datasets. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 479–488, 2010.

D. Cossock and T. Zhang. Statistical analysis of Bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, 2008.

A. Dasgupta and A. Ghosh. Crowdsourced Judgement Elicitation with Endogenous Proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330. ACM, 2013.

S. Davidson, S. Khanna, T. Milo, and S. Roy. Top-k and clustering with noisy comparisons. *ACM Transactions on Database Systems (TODS)*, 39(4):35, 2014.

A. P. Dawid and A. M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied statistics*, pages 20–28, 1979a.

P. A. Dawid and A. M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied statistics*, 28:20–28, 1979b.

B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 395–404, 2005.

B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.*, 33(4):945–964, 2008.

J. DeBoer, G. S. Stump, D. Seaton, and L. Breslow. Diversity in MOOC Students' Backgrounds and Behaviors in Relationship to Performance in 6.002 x. In *Proceedings of the Sixth Learning International Networks Consortium Conference*, volume 4, 2013.

A. Deshpande, L. Hellerstein, and D. Kletenik. Approximation algorithms for stochastic boolean function evaluation and stochastic submodular set cover. In *Proceedings of the* *twenty-fifth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1453–1466. SIAM, 2014.

L. Devroye. The equivalence of weak, strong and complete convergence in l1 for kernel density estimates. *The Annals of Statistics*, pages 896–904, 1983.

L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation.* Springer Science & Business Media, 2012.

P. Diaconis and L. Saloff-Coste. Comparison theorems for reversible Markov chains. *The Annals of Applied Probability*, pages 696–730, 1993.

P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, pages 36–61, 1991.

I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014.

T. Domencich and D. McFadden. *Urban travel demand; a behavioural analysis.* North-Holland, 1975.

J. Duchi, L. Mackey, and M. Jordan. On the consistency of ranking algorithms. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

M. Dudik, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual Dueling Bandits. In *Proceedings of the 28th Conference on Learning Theory*, 2015.

M. Dyer, L. A. Goldberg, M. Jerrum, R. Martin, et al. Markov chain comparison. *Probability Surveys*, 3:89–111, 2006.

E. Emamjomeh-Zadeh, D. Kempe, and V. Singhal. Deterministic and probabilistic binary search in graphs. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 519–532, 2016.

A. Ene and H. L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. *CoRR*, abs/1804.05379, 2018.

H. Esfandiari, A. Karbasi, A. Mehrabian, and V. S. Mirrokni. Regret bounds for batched bandits. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7340–7348. AAAI Press, 2021.

E. Even-Dar, S. Mannor, and Y. Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In *Proceedings of the 15th Conference on Computational Learning Theory*, 2002.

E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

M. Fahrbach, V. S. Mirrokni, and M. Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In T. M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 255–273. SIAM, 2019.

B. Faltings and G. Radanovic. Game theory for data science: Eliciting truthful information. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 11(2):1–151, 2017.

U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

U. Feige, P. Raghavan, D. Peleg, and E. Upfal. Computing with Noisy Information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

A. Fourney, M. Z. Racz, G. Ranade, M. Mobius, and E. Horvitz. Geographic and Temporal Trends in Fake News Consumption During the 2016 US Presidential Election. 2017.

R. Frongillo and I. Kash. Vector-valued property elicitation. In *Proceedings of the 28th Annual Conference on Learning Theory*, 2015.

R. Frongillo and J. Witkowski. A Geometric Perspective on Minimal Peer Prediction. *ACM Transactions on Economics and Computation (TEAC)*, 2017.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS*, 2012.

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*, 20(5):1466–1478, 2012.

A. Gao, J. R. Wright, and K. Leyton-Brown. Incentivizing Evaluation via Limited Access to Ground Truth: Peer-Prediction Makes Things Worse. *EC 2016 Workshop on Algorithmic Game Theory and Data Science*, 2016.

Z. Gao, Y. Han, Z. Ren, and Z. Zhou. Batched multi-armed bandits problem. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 501–511, 2019.

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, pages 359–376, 2011.

D. F. Gleich and L.-h. Lim. Rank aggregation via nuclear norm minimization. In *KDD*, pages 60–68, 2011.

T. Gneiting. Quantiles as optimal point forecasts. *International Journal of Forecasting*, 27 (2):197–207, 2011.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

M. X. Goemans and J. Vondrák. Stochastic covering and adaptivity. In *LATIN 2006: Theoretical Informatics, 7th Latin American Symposium, Valdivia, Chile, March 20-24, 2006, Proceedings*, pages 532–543, 2006.

D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 333–345, 2010.

N. Grammel, L. Hellerstein, D. Kletenik, and P. Lin. Scenario submodular cover. In *Approximation and Online Algorithms - 14th International Workshop, WAOA 2016, Aarhus, Denmark, August 25-26, 2016, Revised Selected Papers*, pages 116–128, 2016.

K. Grant and T. Gneiting. Consistent scoring functions for quantiles. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 163–173. Institute of Mathematical Statistics, 2013.

J. Guiver and E. Snelson. Bayesian inference for Plackett-Luce ranking models. In *ICML*, 2009.

A. Gupta, V. Nagarajan, and S. Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1688–1702, 2017.

R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright. Active Ranking from Pairwise Comparisons and when Parametric Assumptions Dont Help. *arXiv preprint arXiv:1606.08842*, 2016.

L. Hellerstein, D. Kletenik, and P. Lin. Discrete stochastic submodular maximization: Adaptive vs. non-adaptive vs. offline. In *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, pages 235–248, 2015.

E. Hillel, Z. S. Karnin, T. Koren, R. Lempel, and O. Somekh. Distributed exploration in multi-armed bandits. In *NIPS*, 2013.

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 1990.

D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, pages 384–406, 2004.

E. Ie, V. Jain, J. Wang, S. Narvekar, R. Agarwal, R. Wu, H.-T. Cheng, T. Chandra, and C. Boutilier. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. 2019.

S. Im, V. Nagarajan, and R. van der Zwaan. Minimum latency submodular cover. *ACM Trans. Algorithms*, 13(1):13:1–13:28, 2016.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. On Finding the Largest Mean Among Many. *arXiv preprint arXiv:1306.3917v1*, 2013. URL http://arxiv.org/abs/1306.3917.

K. Jamieson, S. Katariya, A. Deshpande, and R. Nowak. Sparse Dueling Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

K. Jamieson, D. Haas, and B. Recht. The Power of Adaptivity in Identifying Statistical Alternatives. In *NIPS*, 2016.

K. G. Jamieson and R. D. Nowak. Active Ranking using Pairwise Comparisons. In *NIPS*, 2011.

M. Jang, S. Kim, C. Suh, and S. Oh. Top-k Ranking from Pairwise Comparisons: When Spectral Ranking is Optimal. *arXiv preprint arXiv:1603.04153*, 2016.

M. Jang, S. Kim, C. Suh, and S. Oh. Optimal sample complexity of m-wise data for top-k ranking. In *NIPS*, 2017.

E. J. Johnson, S. B. Shu, B. G. Dellaert, C. Fox, D. G. Goldstein, G. Häubl, R. P. Larrick, J. W. Payne, E. Peters, D. Schkade, et al. Beyond nudges: Tools of a choice architecture. *Marketing Letters*, 23(2):487–504, 2012.

K.-s. Jun, K. Jamieson, R. Nowak, and X. Zhu. Top Arm Identification in Multi-Armed Bandits with Batch Arm Pulls. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016.

R. Jurca and B. Faltings. Enforcing truthful strategies in incentive compatible reputation mechanisms. In *WINE'05*, volume 3828 LNCS, pages 268–277, 2005.

R. Jurca, B. Faltings, et al. Mechanisms for Making Crowds Truthful. *Journal of Artificial Intelligence Research*, 34(1):209, 2009.

S. Kalyanakrishnan and P. Stone. Efficient Selection of Multiple Bandit Arms: Theory and Practice. In *ICML*, 2010.

S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC Subset Selection in Stochastic

Multi-armed Bandits. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

P. Kambadur, V. Nagarajan, and F. Navidi. Adaptive submodular ranking. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 317–329, 2017.

V. Kamble, D. Marn, N. Shah, A. Parekh, and K. Ramachandran. Truth Serums for Massively Crowdsourced Evaluation Tasks. *The 5th Workshop on Social Computing and User-Generated Content*, 2015.

T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 583–588, 2003.

D. R. Karger, S. Oh, and D. Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *CoRR*, abs/1110.3564, 2011. URL http://arxiv.org/abs/1110.3564.

D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014. doi: 10.1287/opre.2013.1235.

Z. Karnin, T. Koren, and O. Somekh. Almost Optimal Exploration in Multi-Armed Bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

R. M. Karp and R. Kleinberg. Noisy binary search and its applications. In *SODA*, 2007.

E. Kaufmann, O. Cappe, and A. Garivier. On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

A. Khetan and S. Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Annual Conference on Neural Information Processing Systems*, pages 4844–4852, 2016a.

A. Khetan and S. Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *NIPS*, 2016b.

A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First CVPR Workshop on Fine-Grained Visual Categorization*, June 2011.

N. M. Kiefer. Incentive-compatible elicitation of quantiles, 2010. URL https://www.american.edu/cas/economics/info-metrics/pdf/upload/Working-Paper-Kiefer.pdf.

J. Komiyama, J. Honda, H. Kashima, and H. Nakagawa. Regret Lower Bound and Optimal Algorithm in Dueling Bandit Problem. In *Proceedings of the 28th Conference on Learning Theory*, 2015a.

J. Komiyama, J. Honda, and H. Nakagawa. Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015b.

J. Komiyama, J. Honda, and H. Nakagawa. Copeland Dueling Bandit Problem: Regret Lower Bound, Optimal Algorithm, and Computationally Efficient Algorithm. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

J. Konevcny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Y. Kong. Dominantly truthful multi-task peer prediction with a constant number of tasks. In S. Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2398–2411. SIAM, 2020.

Y. Kong and G. Schoenebeck. A Framework For Designing Information Elicitation Mechanism That Rewards Truth-telling. 2016. URL http://arxiv.org/abs/1605.01021.

Y. Kong and G. Schoenebeck. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Trans. Economics and Comput.*, 7 (1):2:1–2:33, 2019.

Y. Kong, K. Ligett, and G. Schoenebeck. Putting Peer Prediction Under the Micro (economic) scope and Making Truth-telling Focal. In *International Conference on Web and Internet Economics*, pages 251–264. Springer, 2016.

F. S. Koppelman and C. Bhat. A self instructing course in mode choice modeling: multinomial and nested logit models. *US Department of Transportation, Federal Transit Administration*, 2006.

C. Kulkarni, K. P. Wei, H. Le, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, and S. R. Klemmer. Peer and Self Assessment in Massive Online Classes. In *Design thinking research*, pages 131–168. Springer, 2015.

B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

N. Lambert and Y. Shoham. Eliciting truthful answers to multiple-choice questions. In *ACM Conference on Electronic Commerce*, 2009.

N. S. Lambert, D. M. Pennock, and Y. Shoham. Eliciting properties of probability distributions. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, 2008.

Y. Lan, J. Guo, X. Cheng, and T.-Y. Liu. Statistical consistency of ranking methods in a rank-differentiable probability space. In *Advances in Neural Information Processing Systems*, 2012.

D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times.* American Mathematical Society, Providence, RI, USA, 2008.

A. Liu, Z. Zhao, C. Liao, P. Lu, and L. Xia. Learning plackett-luce mixtures from partial preferences. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4328–4335. AAAI Press, 2019.

Y. Liu and Y. Chen. Machine-Learning Aided Peer Prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 63–80. ACM, 2017a.

Y. Liu and Y. Chen. Sequential Peer Prediction: Learning to Elicit Effort Using Posted Prices. In *Thirty-First AAAI Conference on Artificial Intelligence*, pages 607–613, 2017b.

Y. Liu and H. Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6226–6236. PMLR, 2020.

Y. Liu and D. P. Helmbold. Online learning using only peer prediction. In S. Chiappa and R. Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2032–2042. PMLR, 2020.

Z. Liu, S. Parthasarathy, A. Ranganathan, and H. Yang. Near-optimal algorithms for shared filter evaluation in data stream systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 133–146, 2008.

D. R. Luce. Individual choice behavior. 1959.

C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

M. L. Malloy, G. Tang, and R. D. Nowak. Quickest search for a rare distribution. In *Information Sciences and Systems (CISS).* IEEE, 2012.

D. Mandal, M. Leifer, D. C. Parkes, G. Pickard, and V. Shnayder. Peer Prediction with Heterogeneous Tasks. *NIPS 2016 Workshop on Crowdsourcing and Machine Learning*, 2016.

S. Mannor and J. N. Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

J. Marschak. Binary choice constraints and random utility indicators. In *Stanford Symposium on Mathematical Methods in the Social Sciences*, pages 312–329, 1960.

L. Maystre and M. Grossglauser. Fast and accurate inference of plackett-luce models. In *NIPS*, 2015.

D. McFadden. *Conditional Logit Analysis of Qualitative Choice Analysis*. New York: Academic Press, 1974.

A. K. Menon and R. C. Williamson. Bipartite ranking: a risk-theoretic perspective. *J. Mach. Learn. Res.*, 17:195:1–195:102, 2016.

M. Mezard and A. Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., New York, NY, USA, 2009.

N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51:1359–1373, 2005.

A. Y. Mitrophanov. Sensitivity and convergence of uniformly ergodic Markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005.

D. Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. *Theory of Computing*, 11:221–235, 2015.

B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden. Active learning for crowd-sourced databases. *CoRR*, abs/1209.3686, 2012.

B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden. Scaling up crowdsourcing to very large datasets: A case for active learning. *PVLDB*, 8(2):125–136, 2014.

F. Nan and V. Saligrama. Comments on the proof of adaptive stochastic set cover based on adaptive submodularity and its implications for the group identification problem in group-based active query selection for rapid diagnosis in time-critical situations. *IEEE Transactions on Information Theory*, 63(11):7612–7614, Nov 2017. ISSN 0018-9448. doi: 10.1109/TIT.2017.2749505.

H. Narasimhan and S. Agarwal. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2913–2921, 2013.

S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *NIPS*, 2012.

S. Negahban, S. Oh, and D. Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017.

S. Oh and D. Shah. Learning mixed multinomial logit model from ordinal data. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 595–603, 2014.

J. Ok, S. Oh, J. Shin, and Y. Yi. Optimality of belief propagation for crowdsourced classification. In *Proc. 33nd Int. Conf. on Machine Learning (ICML)*, pages 535–544, 2016.

S. Parthasarathy. Personal communication. 2018.

V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. Batched bandit problems. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, page 1456, 2015a.

V. Perchet, P. Rigollet, S. Chassang, E. Snowberg, and S. Edu. Batched Bandit Problems. In *COLT*, 2015b.

N. Pippenger. Sorting and selecting in rounds. *SIAM J. Comput.*, 16(6):1032–1038, 1987.

R. L. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

D. Prelec. A Bayesian Truth Serum For Subjective Data. *Science*, 306(5695):462, 2004.

M. Purohit, Z. Svitkina, and R. Kumar. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems*, pages 9661–9670, 2018.

G. Radanovic and B. Faltings. Incentive Schemes for Participatory Sensing. In *AAMAS 2015*, 2015a.

G. Radanovic and B. Faltings. Incentives for Subjective Evaluations with Private Beliefs. In *Proc. 29th AAAI Conf. on Art. Intell. (AAAI'15)*, pages 1014–1020, 2015b.

G. Radanovic and B. Faltings. Incentives for Subjective Evaluations with Private Beliefs. In *Proc. 29th AAAI Conf. on Art. Intell. (AAAI'15)*, pages 1014–1020, 2015c.

G. Radanovic and B. Faltings. Incentive schemes for participatory sensing. In *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1081–1089, 2015d.

G. Radanovic, B. Faltings, and R. Jurca. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4): 48, 2016.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *ICML*, 2014.

S. Ramamohan, A. Rajkumar, and S. Agarwal. Dueling Bandits : Beyond Condorcet Winners to General Tournament Solutions. In *Advances in Neural Information Processing Systems 29*, 2016.

H. G. Ramaswamy and S. Agarwal. Classification calibration dimension for general multiclass losses. In *Advances in Neural Information Processing Systems*, 2012.

H. G. Ramaswamy and S. Agarwal. Convex calibration dimension for multiclass loss matrices. *Journal of Machine Learning Research. To appear*, 2015.

H. G. Ramaswamy, S. Agarwal, and A. Tewari. Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *Advances in Neural Information Processing Systems*, 2013.

P. Ravikumar, A. Tewari, and E. Yang. On NDCG consistency of listwise ranking methods. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

M. D. Reid and R. C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.

T. Rich, K. Hu, and B. Tome. GIFGIF dataset. Data Available: http://www.gif.gf.

Y. Ruan, J. Yang, and Y. Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 74–87, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380539. doi: 10.1145/3406325.3451004. URL https://doi.org/10.1145/3406325.3451004.

P. Rusmevichientong, Z.-J. M. Shen, and D. B. Shmoys. Dynamic Assortment Optimization with a Multinomial Logit Choice Model and Capacity Constraint. *Operations Research*, 58 (6):1666–1680, 2010.

A. Saha and A. Gopalan. Battle of bandits. In *UAI*, pages 805–814, 2018.

A. Saha and A. Gopalan. Combinatorial bandits with relative feedback. In *Advances in Neural Information Processing Systems*, pages 983–993, 2019a.

A. Saha and A. Gopalan. PAC battling bandits in the plackett-luce model. In *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, pages 700–737, 2019b.

D. Sauré and A. Zeevi. Optimal Dynamic Assortment Planning with Demand Learning. *Manufacturing & Service Operations Management*, 15(3):387–404, 2013.

L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

M. J. Schervish. A general method for comparing probability assessors. *The Annals of Statistics*, 17(4):1856–1879, 1989.

M. J. Schervish, J. B. Kadane, and T. Seidenfeld. Characterization of proper and strictly proper scoring rules for quantiles. *Preprint, Carnegie Mellon University*, March 2012.

A. Schuth, H. Oosterhuis, S. Whiteson, and M. de Rijke. Multileave Gradient Descent for Fast Online Learning to Rank. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, 2016.

N. B. Shah and M. J. Wainwright. Simple, Robust and Optimal Ranking from Pairwise Comparisons. *arXiv preprint arXiv:1512.08949*, 2015.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

A. Sheshadri and M. Lease. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proc. 1st AAAI Conf. on Human Computation (HCOMP)*, pages 156–164, 2013.

S. Shetty. Quantifying comedy on YouTube: why the number of o's in your LOL matter. Data Available: https://archive.ics.uci.edu/ml/datasets/YouTube+Comedy+Slam+Preference+Data, 2012.

D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 228–237. IEEE, 2004.

V. Shnayder and D. C. Parkes. Practical Peer Prediction for Peer Assessment. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

V. Shnayder, A. Agarwal, R. Frongillo, and D. C. Parkes. Informed Truthfulness in Multi-Task Peer Prediction. pages 179–196, 2016a.

V. Shnayder, A. Agarwal, R. Frongillo, and D. C. Parkes. Informed Truthfulness in Multi-Task Peer Prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 179–196. ACM, 2016b.

V. Shnayder, R. Frongillo, and D. C. Parkes. Measuring performance of peer prediction mechanisms using replicator dynamics. In *Proc. 25th Int. Joint Conf. on Art. Intell. (IJCAI'16)*, pages 2611–2617, 2016c.

M. Simchowitz, K. Jamieson, and B. Recht. Best-of-K Bandits. In *Proceedings of the 29th Annual Conference on Learning Theory*, 2016.

E. Simpson, S. J. Roberts, I. Psorakis, and A. Smith. Dynamic Bayesian Combination of Multiple Imperfect Classifiers. *Decision Making and Imperfection*, 474:1–35, 2013.

H. A. Soufiani, W. Z. Chen, D. C. Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *NIPS*, 2013.

I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.

I. Steinwart, C. Pasin, R. Williamson, and S. Zhang. Elicitation and identification of properties. In *Proceedings of the 27th Annual Conference on Learning Theory*, 2014.

Y. Sui, V. Zhuang, J. W. Burdick, and Y. Yue. Multi-dueling Bandits with Dependent Arms. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

C. Swamy and D. B. Shmoys. Sampling-based approximation algorithms for multistage stochastic optimization. *SIAM Journal on Computing*, 41(4):975–1004, 2012.

A. Tewari and P. L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

K. E. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2003.

T. Urvoy, F. Clerot, R. Feraud, and S. Naamane. Generic Exploration and K-armed Voting Bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

L. G. Valiant. Parallelism in comparison problems. *SIAM J. Comput.*, 4(3):348–355, 1975.

E. Vernet, R. C. Williamson, and M. D. Reid. Composite multiclass losses. In *Advances in Neural Information Processing Systems*, 2011.

J. Wilkowski, A. Deutsch, and D. M. Russell. Student Skill and Goal Achievement in the Mapping with Google MOOC. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 3–10. ACM, 2014.

R. C. Williamson, E. Vernet, and M. D. Reid. Composite multiclass losses. *Journal of Machine Learning Research*, 17(222):1–52, 2016.

J. Witkowski and D. C. Parkes. Learning the Prior in Minimal Peer Prediction. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content at the ACM Conference on Electronic Commerce*, page 14, 2013.

J. Witkowski, Y. Bachrach, P. Key, and D. Parkes. Dwelling on the negative: Incentivizing effort in peer prediction. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

H. Wu and X. Liu. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 649–657, 2016.

F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

Y. Yamaguchi and T. Maehara. Stochastic packing integer programs with few queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 293–310, 2018.

A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213, 1979.

Y. Yue and T. Joachims. Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Y. Yue and T. Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The K-armed Dueling Bandits Problem. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012.

M. B. Zafar, K. P. Gummadi, and C. Danescu-Niculescu-Mizil. Message Impartiality in Social Media Discussions. In *ICWSM*, pages 466–475, 2016.

H. Zhang, Y. Ma, and M. Sugiyama. Bandit-based task assignment for heterogeneous crowdsourcing. *Neural Comput.*, 27(11):2447–2475, 2015. doi: 10.1162/NECO_a_00782. URL https://doi.org/10.1162/NECO_a_00782.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–134, 2004a.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004b.

Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. *Journal of Machine Learning Research*, 17(102): 1–44, 2016.

Z. Zhao and L. Xia. Learning mixtures of plackett-luce models from structured partial orders. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10143–10153, 2019.

D. Zhou, Q. Liu, J. C. Platt, C. Meek, and N. B. Shah. Regularized minimax conditional entropy for crowdsourcing. *CoRR*, abs/1503.07240, 2015. URL http://arxiv.org/abs/1503.07240.

M. Zoghi, S. Whiteson, R. Munos, and M. de Rijke. Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

M. Zoghi, Z. Karnin, S. Whiteson, and M. de Rijke. Copeland Dueling Bandits. In *Advances in Neural Information Processing Systems 28*, 2015a.

M. Zoghi, S. Whiteson, and M. de Rijke. MergeRUCB: A method for large-scale online ranker evaluation. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, 2015b.

M. Zoghi, T. Tunys, M. Ghavamzadeh, B. Kveton, C. Szepesvari, and Z. Wen. Online learning to rank in stochastic click models. In *ICML*, pages 4199–4208, 2017.