Algorithms in Strategic or Noisy Environments

Jieming Mao

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Mark Braverman

September 2018

Abstract

Algorithms are often used to interact with agents. When the input is collected from agents instead of being directly given, designing algorithms becomes more challenging. Two main challenges arise here: (i) agents may lie to maximize their own utility functions and we need to take their incentives into account (ii) the uncertainty in agents' behavior makes the input appear to be noisy.

In this thesis, we study these two challenges in several contexts: the multi-armed bandit problem, combinatorial auctions and rank aggregation. Our goal is to understand how these strategic and noisy factors make the problem harder and to design new techniques which make algorithms robust against these factors.

In Part I (Chapter 2 and Chapter 3), we study multi-armed bandit algorithms in strategic environments where rewards are decided by actions taken by strategic agents. We show that traditional multi-armed bandit algorithms could fail and we also develop multi-armed bandit algorithms which achieve good performance in strategic envrionments.

In Part II (Chapter 4 and Chapter 5), we focus on combinatorial auctions which are a natural testbed for truthful mechanisms. We characterize the power of truthful mechanisms in several settings and make progress in understanding whether truthful mechanisms are as powerful as algorithms.

In Part III (Chapter 6, Chapter 7 and Chapter 8), we study the top-k ranking problem. In this problem, even if the agents are perfectly incentivized, their reported comparison results could still be noisy caused by reasons like limit of knowledge and subjective preferences. We deisgn algorithms which aggregate noisy comparisons to output the set of top items.

Acknowledgements

Foremost, I would like to thank my advisor Mark Braverman for his support over the years. He has always been a great role model for me and his guidance helped me to mature as a researcher. There were numerous times when I got stuck in research, the inspiring discussions with him led to new progress and finally many results of this thesis.

I am grateful to Matt Weinberg for his guidance, encouragement and collaboration. His advice on both research as well as on my career has been invaluable.

I would like to thank Bernard Chazelle, Elad Hazan and Ran Raz who devoted time to serve on my thesis committee.

I would like to thank my collaborators over the past few years: Mark Braverman, Xi Chen, Ankit Garg, Ran Gelles, Sivakanth Gopi, Nicole Immorlica, Young Kun Ko, Yuanzhi Li, Brendan Lucier, Rafail Ostrovsky, Jon Schneider, Vasilis Syrgkanis, Christos Tzamos, Dave Touchette and Matt Weinberg.

I would like to thank CS department at Princeton, especially Mitra Kelly and Nicki Gotsis for their administrative work over the years. I would like to thank grants and fellowships: CCF-1215990, CCF-1149888, CCF-1525342, School of Engineering and Applied Science Award for Excellence and Siebel Scholarship.

I would like to thank my friends, my fellow graduate students especially those people who have been my officemates: Ankit Garg, Sumegha Garg, Sivakanth Gopi, Young Kun Ko, Yuanzhi Li, Rafael Oliveira, Jon, Schneider and Ariel Schvartzman. They made my graduate student daily life much more interesting.

Last but not least, I would like to thank my parents Long and Jinling for their love and support throughout the past 27 years.

To my parents, Long and Jinling.

Contents

Chapter 1

Introduction

Traditional algorithms are designed to efficiently compute outputs when inputs are directly given. In recent years, motivated by various applications on the Internet, there has been a lot of interests understanding the performance of algorithms whose inputs are collected from agents. For example, in crowdsourcing, data are collected from workers on crowdsourcing platforms and later used by algorithms. As another example, in sponsered search auctions, auctioneers' algorithms decide how the advertisement slots are assigned based on bidders' bids. In these examples, it is clear that we should reason about how agents provide inputs. While doing so, two main challenges arise.

The first challenge is that, algorithms do not have the power to directly enforce agents to provide the needed inputs. Instead, agents may strategically provide inputs which maximize their own utility functions. In the example of sponsored search auctions, bidders don't necessarily need to bid their true values. In order to reason about such incentive issues, tools from game theory and mechanism design should be used in the algorithm design.

The second chanllenge is that, even when agents are perfectly incentivized to provide their information in a certain way, algorithms still need to face some extra uncertainty caused by the heterogeneity among agents and noise generated in the interaction between algorithms and agents. In the example of crowdsourcing, workers on the crowdsourcing platforms may have subjective perferences and their knowledge sometimes is not sufficient for the tasks. As a result, data collected from crowdsourcing can be noisy. Algorithms have to extract useful information from the noisy inputs and solve algorithmic tasks.

In this thesis, we study these two challenges in the contexts of multi-armed bandit, combinatorial auctions and rank aggregation. These problems are much better understood in the absence of strategic or noisy factors. We focus on understanding how much more difficult these problems become when inputs are strategic or noisy and develop tools for algorithms in strategic or noisy environments.

1.1 Overview

In section, we give an overview of our results.

1.1.1 Multi-armed Bandit with Strategic Inputs

Multi-armed bandit is a fundamental decision problem in machine learning that models the tradeoff between exploration and exploitation in an online setting. In the classic multi-armed bandit problem, an algorithm A chooses one of K arms per round, over T rounds. On round t, the algorithm receives some reward $X_{i,t} \in [0,1]$ for pulling arm i. Let I_t denote the arm pulled by the algorithm A at round t. The *regret* of A is defined as

$$\mathsf{Reg}(A) = \max_{i \in \{1,\ldots,K\}} \sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{I_t,t}.$$

Good regret bounds have been achieved if the rewards $X_{i,t}$'s are drawn stochastically or chosen by a non-adaptive (oblivious) adversary. If adversary decides rewards adaptively according to the history, the regret notion needs to be refined to incorporate the adaptiveness. This is addressed by the notion "policy regret" [13]. [13] showed that no algorithms can achieve sublinear policy regret in the most general setting.

When multi-armed bandit algorithms are used in strategic environments, the rewards $X_{i,t}$'s are decided by actions taken by strategic agents who may use adaptive strategies. So the rewards cannot be simply considered as stochastic or decided by a non-adaptive adversary. The algorithms with good regret bounds don't simply apply here. On the other hand, strategic agents are not completely adversarial. They want to maximize their own utilities. So there is still hope to achieve good policy regret bounds. In Part I (Chapter 2 and Chapter 3), we study the performance of multi-armed bandit algorithms in two different strategic settings.

In Chapter 2, we study the setting where a single seller repeatedly sells a single item to a single buyer. The buyer is using a no-regret learning algorithm to make online decisions. This no-regret learning algorithm is interacting with a strategic agent — the seller who strategically maximize its own revenue. We ask the following question: if the seller knows the buyer is no-regret learning over time, can the seller achieve average revenue more than the Myerson revenue? Here Myerson revenue is the maximum possible revenue of selling a single item in a single round.

This problem can be considered as a special case of a two-player repeated game where one player is using no-regret learning and the other player is strategically maximize its own utility. In our specific setting, we manage to chracterize the maximum utility of the strategic player (the seller). Interestingly, this number highly depends on which no-regret learning algorithm is used.

We show that if the buyer is using EXP3 (or any other "mean-based" learning algorithm[1]), the seller can achieve average revenue higher than the Myerson revenue. In fact, it can be arbitrarily close to the expected buyer's value of the item. Even when the seller is restricted to auction format where overbidding is always dominated, we show that the optimal revenue can still be better than the Myerson revenue. We characterize the seller's

[1] Informally, a "mean-based" learning algorithm is define as a learning algorithm which chooses arms of higher means with much higher probabilities. This notion is formally defined in Chapter 2.

optimal revenue by a linear program and show that seller's optimal strategy is a pay-your-bid format with decreasing reserve over time.

On the other hand, if the buyer is using some sophisticated no-regret learning algorithm taylored for this problem, we show that the seller cannot get average revenue more than the Myerson revenue. We develop a no-regret learning algorithm for the buyer such that no seller's strategy can get more than the Myerson revenue. In this case, the optimal strategy for the seller is just to post the Myerson reserve every round.

In Chapter 3, we study a different strategic environment for multi-armed bandit algorithms. In this setting, each arm corresponds to a strategic agent and the principal pulls one arm each round. When an arm gets pulled, it receives some private saving v_a and can choose amount x_a to pass on to the principal (keeping $v_a - x_a$ for itself).

We first show that algorithms that perform well in the classic adversarial multi-armed bandit setting necessarily perform poorly: For all algorithms that guarantee low regret in an adversarial setting, there exist distributions and an $o(T)$-approximate Nash equilibrium for the arms where the principal receives reward $o(T)$.

On the other hand, we show that there exists an algorithm for the principal that induces a game among the arms where each arm has a dominant strategy. Moreover, for every $o(T)$-approximate Nash equilibrium, the principal receives expected reward $\mu' T - o(T)$, where μ' is the second-largest of the means of arms' private rewards. This algorithm maintains its guarantee if the arms are non-strategic ($x_a = v_a$), and also if there is a mix of strategic and non-strategic arms.

1.1.2 Truthful Mechanisms in Combinatorial Auctions

In a combinatorial auction, a designer with m items wishes to allocate them to n bidders so as to maximize the *social welfare*. That is, if bidder i has a monotone valuation function $v_i : 2^{[m]} \to \mathbb{R}_+$, the designer wishes to find disjoint sets $S_1, \ldots, S_n$ maximizing $\sum_i v_i(S_i)$. The history of combinatorial auctions is rich, and the problem has been considered with and without incentives, with and without Bayesian priors, and in various models of computation. The overarching theme in all of these works is to try and answer the following core question: *Are truthful mechanisms as powerful as (not necessarily truthful) algorithms*?

For many instantiations of the above question, the answer is surprisingly yes. For example, without concern for computational/communication complexity, the celebrated Vickrey-Clarke-Groves auction is a truthful mechanism that always selects the welfare-maximizing allocation (and therefore achieves welfare equal to that of the best algorithm) [187, 63, 110]. Of course, the welfare maximization problem is NP-hard and also requires exponential communication between the bidders, even to guarantee a $1/\sqrt{m}$-approximation. A poly-time algorithm (with polynomial communication) is known to match this guarantee [168, 131, 43], and interestingly, a poly-time truthful mechanism (with polynomial communication) was later discovered as well [139].

The state of affairs gets even more interesting if we restrict to proper subclasses of monotone valuations such as *submodular* valuations.[2] Here, a very simple greedy algorithm is known to find a 1/2-approximation in both poly(n, m) black-box value queries to each $v_i(\cdot)$, and polynomial runtime (in n, m, and the description complexity of each $v_i(\cdot)$) [140], and a series of improvements provides now a (1-1/e)-approximation, which is tight [188, 153, 83]. Yet, another series of works also proves that any truthful mechanism that runs in polynomial time (in n, m, and the description complexity of each $v_i(\cdot)$), or makes only poly(n, m) black-box value queries to each $v_i(\cdot)$ achieves at best an $1/m^{\Omega(1)}$-approximation [67, 75, 87, 84]. So while poly-time algorithms, or algorithms making poly(n, m) black-box value queries can achieve constant-factor approximations, poly-time truthful mechanisms and truthful mechanisms making poly(n, m) black-box value queries can only guarantee an $1/m^{\Omega(1)}$-approximation, and there is a separation.

In Part II (Chapter 4 and Chapter 5), we study the power of truthful mechanisms in two less well-understood settings.

[2] A function is submodular if $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$.

In Chapter 4, we focus on truthful mechanisms with polynomial communication. It is motivated by posted-price mechanisms which are truthful mechanisms with polynomial many demand queries. The hardness results discussed above don't simply apply to posted-price mechanisms since a demand query is NP-hard or requires exponential many value queries to answer. However, posted-price mechanisms are considered "natural" and their approximation ratios have been subsequently improved to $1/O(\sqrt{\log m})$ [81, 74, 134, 76] for XOS valuations[3]. No separation between polytime algorithms and posted-price mechanisms is known. Recently, [77] provided a clear path to possibly proving a separation. The paper shows that for XOS valuations, a posted-price mechanism for two bidders implies a simultaneous communication protocol with polynomial communication cost and therefore lower bounds on simultaneous communication complexity would imply lower bounds on posted-price mechanisms. We follow this direction to study the simultaneous communication complexity of two-player combinatorial auctions. We give a tight characterization for the binary XOS valuation class (a special case of XOS) and also extensions for general XOS valuations.

In Chapter 5, we study the power of interpolation mechanisms that interpolate between non-truthful and truthful protocols. Specifically, an interpolation mechanism has two phases. In the first phase, the bidders participate in some non-truthful protocol whose output is itself a truthful protocol. In the second phase, the bidders participate in the truthful protocol selected during phase one. Note that virtually all existing auctions have either a non-existent first phase (and are therefore truthful mechanisms), or a non-existent second phase (and are therefore just traditional protocols, analyzed via the Price of Anarchy/Stability).

The goal of this chapter is to understand the benefits of interpolation mechanisms versus truthful mechanisms or traditional protocols, and develop the necessary tools to formally study them. Interestingly, we exhibit settings where interpolation mechanisms greatly outperform the optimal traditional and truthful protocols. Yet, we also exhibit settings where interpolation mechanisms are provably no better than truthful ones. Finally, we apply our

[3] A function is XOS if it is the maximum of additive functions.

new machinery to prove that the single-bid mechanism of Devanur et. al. [70] achieves the optimal price of anarchy among a wide class of protocols.

1.1.3 Rank Aggregation with Noisy Comparisons

Rank aggregation is an important problem in machine learning and it finds many applications in crowdsourcing, recommendation systems, peer grading and web search. We focus on the problem called top-k (or partition): Given a set of n items with an unknown underlying order, output the set of top-k items out of n items using comparison results collected from agents. In these applications, the noise in the communication and the subjective preferences among agents make the comparison results noisy. In Part III (Chapter 6, Chapter 7 and Chapter 8), we study several interesting aspects of the top-k problem with noisy comparisons.

In Chapter 6, we study the round complexity of top-k algorithms. In many of these applications, multiple rounds of interaction are costly. For example, in peer grading, each round of interaction might take a week. We study the tradeoff between the round complexity (the adaptiveness of algorithms) and the sample complexity (how many samples are needed) of the top-k ranking problem in standard noise models[4]. We first show that one-round algorithms have much worse performance than fully adaptive algorithms. Then we tightly characterize the minimum number of rounds an algorithm requires to achieve the performance of fully adaptive algorithms. Surprisingly, this number is quite small: 4 and $\Theta(\log^*(n))$ in the noise models we study, despite the poor performance of one-round algorithms.

In Chapter 7, we study a more general noise model named strong stochastic transitivity (SST) noise model. SST requires that for any items i, j, l, the probability we observe item i beats item l in the comparisons is at least the probability we observe item j beats item l. In this chapter, we present an algorithm which has a competitive ratio of $\tilde{O}(\sqrt{n})$; i.e. to solve any instance of the top-k problem, our algorithm needs at most $\tilde{O}(\sqrt{n})$ times as many samples needed as the best possible algorithm for that instance. In contrast, all previous

[4]Two noise models we consider here are: erasure (each comparison is erased with probability $1-\gamma$) and noisy (each comparison is correct with probability $1/2+\gamma/2$ and incorrect otherwise.

known algorithms for the top-k problem have competitive ratios of $\tilde{\Omega}(n)$ or worse. We further show that this is tight up to polylogarithmic factors: any algorithm for the top-k problem has competitive ratio at least $\tilde{\Omega}(\sqrt{n})$. Our result is very related to the notion "instance optimal" [96] and it directly implies that there are no instance optimal algorithms under the SST noise model.

In Chapter 8, we study the top-k problem under the multinomial logit (MNL) model to understand how much more helpful multi-wise comparisons are. In MNL, a multi-wise comparison within set S will return item i with probability $\frac{\theta_i}{\sum_{j \in S} \theta_j}$ for $i \in S$. Here θ_i is the underlying preference score of item i. We design a new active ranking algorithm without using any information about the underlying items' preference scores. We also establish a matching lower bound on the sample complexity even when the set of preference scores is given to the algorithm. These two results together show that the proposed algorithm is nearly instance optimal. With this tight sample complexity bound, we also characterize when multi-wise comparisons are more helpful than pairwise comparisons.

1.2 Preliminaries

In this section, we describe some tools we use throughout the thesis.

1.2.1 Game Theory

Here we review some standard definitions from game theory which are used in Part I and Part II.

Consider a finite game with n players. Player i has a set of strategies S_i and it can choose a mixed strategy $x_i \in \Delta^{S_i}$ (a distribution over the strategy set S_i). For a strategy profile $s = (s_1, ..., s_n)$, player i's utility is defined as $u_i(s)$. For a mixed strategy profile $x = (x_1, ..., x_n)$, player i's expected utility is defined as $u_i(x) = \mathrm{E}_{s \sim x}[u_i(s)]$. We use notation x_{-i} to denote the mixed strategies of all players except player i.

Definition 1.2.1 (Nash Equilibrium)**.** *A mixed strategy profile x is a Nash equilbrium if for all $i \in [n]$ and $s_i \in S_i$,*

$$u_i(x) \geq u_i(s_i, x_{-i}).$$

Definition 1.2.2 (ε-Nash Equilibrium)**.** *A mixed strategy profile x is an ε-Nash equilbrium if for all $i \in [n]$ and $s_i \in S_i$,*

$$u_i(x) \geq u_i(s_i, x_{-i}) - \varepsilon.$$

Definition 1.2.3 (Dominant Strategy)**.** *We say that strategy x_i is a dominant strategy for player i, if for all x_{-i} and $s_i \in S_i$,*

$$u_i(x_i, x_{-i}) \geq u_i(s_i, x_{-i}).$$

1.2.2 Information Theory

We briefly review some standard facts and definitions from information theory which is used in Chapter 4, Chapter 6 and Chapter 7. For a more detailed introduction, we refer the reader to [66].

Throughout this thesis, we use log to refer to the base 2 logarithm and use ln to refer to the natural logarithm.

Definition 1.2.4. *The* entropy *of a random variable X, denoted by $H(X)$, is defined as $H(X) = \sum_x \Pr[X = x] \log(1/\Pr[X = x])$.*

If X is drawn from Bernoulli distributions $\mathcal{B}(p)$, we use $H(p) = -(p \log p + (1-p)(\log(1-p))$ to denote $H(X)$.

Definition 1.2.5. *The* conditional entropy *of random variable X conditioned on random variable Y is defined as $H(X|Y) = \mathbb{E}_y[H(X|Y = y)]$.*

Fact 1.2.1. $H(XY) = H(X) + H(Y|X)$.

Definition 1.2.6. *The* mutual information *between two random variables* X *and* Y *is defined as* $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$.

Definition 1.2.7. *The* conditional mutual information *between* X *and* Y *given* Z *is defined as* $I(X;Y|Z) = H(X|Z) - H(X|YZ) = H(Y|Z) - H(Y|XZ)$.

Fact 1.2.2. *Let* X_1, X_2, Y, Z *be random variables, we have* $I(X_1X_2;Y|Z) = I(X_1;Y|Z) + I(X_2;Y|X_1Z)$.

Fact 1.2.3. *Let* X, Y, Z, W *be random variables. If* $I(Y;W|X,Z) = 0$*, then* $I(X;Y|Z) \geq I(X;Y|ZW)$.

Fact 1.2.4. *Let* X, Y, Z, W *be random variables. If* $I(Y;W|Z) = 0$*, then* $I(X;Y|Z) \leq I(X;Y|ZW)$.

Definition 1.2.8. *The* Kullback-Leibler divergence *between two random variables* X *and* Y *is defined as* $D(X\|Y) = \sum_x \Pr[X = x] \log(\Pr[X = x]/\Pr[Y = x])$.

If X and Y are drawn from Bernoulli distribution B_p and B_q, we write $D(p\|q)$ as an abbreviation for $D(X\|Y)$.

Fact 1.2.5. *Let* X, Y, Z *be random variables, we have* $I(X;Y|Z) = \mathbb{E}_{x,z}[D((Y|X = x, Z = z)\|(Y|Z = z))]$.

Fact 1.2.6. *Let* X, Y *be random variables,*

$$\sum_x \frac{|\Pr[X = x] - \Pr[Y = x]|^2}{2\max\{\Pr[X = x], \Pr[Y = x]\}} \leq \ln(2) \cdot D(X\|Y) \leq \sum_x \frac{|\Pr[X = x] - \Pr[Y = x]|^2}{\Pr[Y = x]}.$$

Proof. For notation convenience, let $p(x) = Pr[X = x]$ and $q(x) = Pr[Y = x]$. Let's first prove the right-hand side.

$$\begin{aligned}
\ln(2) \cdot D(X\|Y) &= \sum_x p(x) \ln\left(\frac{p(x)}{q(x)}\right) \\
&\leq \ln\left(\sum_x \frac{p(x)^2}{q(x)}\right) \quad \text{(by concavity of } \ln(z)) \\
&\leq \sum_x \frac{p(x)^2}{q(x)} - 1 \\
&= \sum_x \frac{(p(x) - q(x))^2}{q(x)}.
\end{aligned}$$

For the left-hand side, consider any convex function f such that $f''(x) \geq m > 0$ for all $x \in [a, b]$. By strong convexity, for $x, y \in [a, b]$, we have

$$f(y) \geq f(x) + f'(x)(y - x) + \frac{m(y - x)^2}{2}.$$

Let $f(x) = x \ln x$. For $x \in [a, b]$, we have $f''(x) \geq \frac{1}{b}$. Therefore,

$$a \ln a \geq b \ln b + (a - b)(1 + \ln b) + \frac{(a - b)^2}{2b}.$$

and then

$$a \ln\left(\frac{a}{b}\right) \geq (a - b) + \frac{(a - b)^2}{2b}.$$

Similarly, we have

$$b \ln\left(\frac{b}{a}\right) \geq (b - a) + \frac{(a - b)^2}{2b}.$$

Thus

$$\begin{aligned}
\ln(2) \cdot D(X\|Y) &= \sum_x p(x) \ln\left(\frac{p(x)}{q(x)}\right) \\
&\geq \sum_x \left[p(x) - q(x) + \frac{(p(x)-q(x))^2}{2\max\{p(x), q(x)\}}\right] \\
&= \sum_x \frac{(p(x)-q(x))^2}{2\max\{p(x), q(x)\}}.
\end{aligned}$$

□

1.2.3 Communication Complexity

The two-party communication model was introduced by Yao [190]. Here we review some definitions and notations in communication complexity which are used in Chapter 4 and Chapter 5. For a more detailed introduction, see [135].

In the two-party communication model, Alice and Bob want to jointly compute a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Alice is only given input $x \in \mathcal{X}$ and Bob is only given input $y \in \mathcal{Y}$. In order to compute function f, they have to communicate with each other following a protocol π which specifies when the communication is over, who sends the next bit if the communication is not over, and the function of each transmitted bit given the history, the input of the person who sends this bit and the shared randomness. The transcript of a protocol is a concatenation of all bits exchanged. The communication cost of a communication protocol π is define as the maxmimum number of bits exchanged on the worst input.

Definition 1.2.9 (Deterministic Communication Complexity)**.** *For a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, the deterministic communication complexity of f is defined as the communication cost of the best deterministic protocol for computing f.*

Definition 1.2.10 (Randomized Communication Complexity)**.** *For a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and a parameter $\varepsilon > 0$, the randomized communication complexity of f denotes the* *communication cost of the best randomized protocol for computing f with error at most ε on every input.*

The two-party communication model can be generalized to the multi-party number-in-hand model. In this model, there are k players, each with a private input X_i and players wish to compute a joint function $f(X_1, ..., X_k)$. In Chapter 4 and Chapter 5, we focus on the communication model in the context of combinatorial auctions. This model is a special case of the multi-party number-in-hand model. The players are the bidders and the auctioneer. Each bidder's input is its valuation function and the goal is to output the optimal allocation of items.

Part I

Multi-armed Bandit Problems with Strategic Inputs

Chapter 2

Selling to a No-Regret Buyer

The results of this chapter are based on joint work with Mark Braverman, Jon Schneider and Matt Weinberg [37].

2.1 Introduction

Consider a bidder trying to decide how much to bid in an auction (for example, a sponsored search auction). If the auction happens to be the truthful Vickrey-Clarke-Groves auction [187, 63, 110], then the bidder's decision is easy: simply bid your value. If instead, the bidder is participating in a Generalized First-Price (GFP) or Generalized Second-Price (GSP) auction, the optimal strategy is less clear. Bidders can certainly attempt to compute a Bayes-Nash equilibrium of the associated game and play accordingly, but this is unrealistic due to the need for accurate priors and extensive computation.

Alternatively, the bidders may try to learn a best-response over time (possibly offloading the learning to commercial bid optimizers). We specifically consider bidders who *no-regret learn*, as empirical work of [160] shows that bidder behavior on Bing is largely consistent with no-regret learning (i.e. for most bidders, there exists a per-click value such that their behavior guarantees no-regret for this value). From the perspective of a revenue-maximizing auction designer, this motivates the following question: **If a seller knows that buyers are no-regret learning over time, how should they maximize revenue?**

This question is already quite interesting even when there is just a single item for sale to a single buyer. We consider a model where in every round t, the seller solicits a bid $b_t \in [0,1]$ from the buyer, then allocates the item according to some allocation rule $x_t(\cdot)$ and charges the bidder according to some pricing rule $p_t(\cdot)$ (satisfying $p_t(b) \leq b \cdot x_t(b)$ for all t, b).[1] Note that the allocation and pricing rules (henceforth, auction) can differ from round to round, and that the auction need not be truthful. Each round, the bidder has a value v_t drawn independently from $\mathcal{D}$, and uses some no-regret learning algorithm to decide which bid to place in round t, based on the outcomes in rounds $1, \ldots, t-1$ (we will make clear exactly what it means for a buyer with changing valuation to play no-regret in Section 2.2, but one can think of v_t as providing a "context" for the bidder during round t). The same mathematical model can also represent a population $\mathcal{D}$ of many indistinguishable buyers with fixed values who each separately no-regret learn - see Section 2.2.3 for further details.

One default strategy for the seller is to simply to set Myerson's revenue-optimal reserve price for $\mathcal{D}$, $r(\mathcal{D})$, in every round (that is, $x_t(b_t) = I(b_t \geq r(\mathcal{D}))$, $p_t(b_t) = r(\mathcal{D}) \cdot I(b_t \geq r(\mathcal{D}))$ for all t, where $I(\cdot)$ is the indicator function). It's not hard to see that *any* no-regret learning algorithm will eventually learn to submit a winning bid during all rounds where $v_t > r(\mathcal{D})$, and a losing bid whenever $v_t < r(\mathcal{D})$. Note that this observation appeals only to the fact that the buyer guarantees no-regret, and makes no reference to any specific algorithm the buyer might use. So if $\mathsf{Rev}(\mathcal{D})$ denotes the expected revenue of the optimal reserve price when a single buyer is drawn from $\mathcal{D}$, the default strategy guarantees the seller revenue $T \cdot \mathsf{Rev}(\mathcal{D}) - o(T)$ over T rounds. The question then becomes whether or not the seller can beat this benchmark, and if so by how much.

The answer to this question isn't a clear-cut yes or no, so let's start with the following instantiation: how much revenue can the seller extract if the buyer runs EXP3 [18]? In

[1] Of course, the pricing rule can be implemented by charging $p_t(b)/x_t(b)$ whenever the item is awarded if ex-post individual rationality is desired.

Theorem 2.3.1, we show that the seller can actually do *much* better than the default strategy: it's possible to extract revenue per round equal to (almost) the full expected welfare! That is, if $\mathsf{Val}(\mathcal{D}) = \mathbb{E}_{v \leftarrow \mathcal{D}}[v]$, there exists an auction that extracts revenue $T \cdot \mathsf{Val}(\mathcal{D}) - o(T)$ for all $\mathcal{D}$.[2] It turns out this result holds not only for EXP3, but for any learning algorithm with the following (roughly stated) property: if at time t, the mean reward of action a is significantly larger than the mean reward of action b, the learning algorithm will choose action b with negligible probability. We call a learning algorithm with this property a "mean-based" learning algorithm and note that many commonly used learning algorithms - EXP3, Multiplicative Weights Update [14], and Follow-the-Perturbed-Leader [114, 128, 129] - are 'mean-based' (see Section 2.2 for a formal definition).

We postpone all intuition until Section 2.3.1 with a worked-through example, but just note here that the auction format is quite unnatural: it "lures" the bidder into submitting high bids early on by giving away the item for free, and then charging very high prices (but still bounded in $[0, 1]$) near the end. The transition from "free" to "high-price" is carefully coordinated across different bids to achieve the revenue guarantee.

This result motivates two further directions. First, do there exist other no-regret algorithms for which full surplus extraction is impossible for the seller? In Theorem 2.3.2, we show that the answer is yes. In fact, there is a simple no-regret algorithm $\mathcal{A}$, such that when the bidder uses algorithm $\mathcal{A}$ to bid, the default strategy (set the Myerson reserve every round) is optimal for the seller. We again postpone a formal statement and intuition to Section 2.3.2, but just note here that the algorithm is a natural adaptation of EXP3 (or in fact, any existing no-regret algorithm) to our setting.

Finally, it is reasonable to expect that bidders might use off-the-shelf no-regret learning algorithms like EXP3, so it is still important to understand what the seller can hope to achieve if the buyer is specifically using such a "mean-based" algorithm (formal definition in Section 2.2). Theorem 2.3.1 is perhaps unsatisfying in this regard because the proposed

[2] The order of quantifiers in this sentence is correct: it is actually the same auction format that works for all $\mathcal{D}$.

auction is so unnatural. It turns out that the key property separating natural untruthful auctions (e.g. GSP/GFP) from the unnatural auction above is whether overbidding is a dominated strategy. That is, in our unnatural auction, if the bidder truly hopes to guarantee low regret they must seriously consider overbidding (and this is how the auction lures them into bidding way above their value). In both GSP and GFP, overbidding is dominated, so the bidder can guarantee no regret while overbidding with probability 0 in every round.

The final question we ask is the following: if the buyer is using EXP3 (or any "mean-based" algorithm), never overbids (we call such a bidder *conservative*), how much revenue can the seller extract using an auction where overbidding is dominated in every round? It turns out that the auctioneer can still outperform the default strategy, but not extract full welfare. Instead, we identify a linear program (as a function of $\mathcal{D}$) that tightly characterizes the optimal revenue the seller can achieve in this setting when the buyer's values are drawn from $\mathcal{D}$. Moreover, we show that the auction that achieves this guarantee is natural, and can be thought of as a pay-your-bid auction with decreasing reserves over time. Finally, we show that this "mean-based revenue" benchmark, $\mathsf{MBRev}(\mathcal{D})$ lies truly in between the Myerson revenue and the expected welfare: for all c, there exists a distribution $\mathcal{D}$ over values such that $c \cdot T \cdot \mathsf{Rev}(\mathcal{D}) < \mathsf{MBRev}(\mathcal{D}) < \frac{1}{c} \cdot T \cdot \mathsf{Val}(\mathcal{D})$. In other words, the seller's mean-based revenue may be unboundedly better than the default strategy, yet simultaneously unboundedly far from the expected welfare. We provide formal statements and a detailed proof overview of these results in Section 2.3.3. To briefly recap, our main results are the following:

1. If the buyer uses a "mean-based" learning algorithm like EXP3, the seller can extract revenue $(1 - \varepsilon)T \cdot \mathsf{Val}(\mathcal{D}) - o(T)$ for any constant $\varepsilon > 0$ (Theorem 2.3.1).

2. There exists a natural no-regret algorithm $\mathcal{A}$ such that when the buyer bids according to $\mathcal{A}$, the seller's default strategy (charging the Myerson reserve every round) is optimal (Theorem 2.3.2).

3. If the buyer uses a "mean-based" algorithm only over undominated strategies, the seller can extract revenue $\mathsf{MBRev}(\mathcal{D})$ using an auction where overbidding is dominated in every round. Moreover, we characterize $\mathsf{MBRev}(\mathcal{D})$ as the value of a linear program, and show it can be simultaneously unboundedly better than $T \cdot \mathsf{Rev}(\mathcal{D})$ and unboundedly worse than $T \cdot \mathsf{Val}(\mathcal{D})$ (Theorems 2.3.4, 2.3.3 and 2.3.5).

Our plan for the remaining sections is as follows. Below, we overview our connection to related work. Section 2.2 formally defines our model. Section 2.3 works through a concrete example, providing intuition for all three results. Section 2.4 discusses conclusions and open problems.

2.1.1 Related Work

There are two lines of work that are most related to ours. The first is that of *dynamic auctions*, such as [164, 15, 151, 152, 143]. Like our model, there are T rounds where the seller has a single item for sale to a single buyer, whose value is drawn from some distribution every round. However, the buyer is fully strategic and processes fully how their choices today affect the seller's decisions tomorrow (e.g. they engage with deals of the form "pay today to get the item tomorrow"). Additional closely related work is that of Devanur et al. studying the Fishmonger problem [73, 122]. Here, there is again a single buyer and seller, and T rounds of sale. Unlike our model, the buyer draws a value from $\mathbb{D}$ once during round 0 and that value is fixed through all T rounds (so the seller could try to learn the buyer's value over time). Also unlike our model, they study perfect Bayesian equilibria (where again the buyer is fully strategic, and reasons about how their actions today affect the seller's behavior tomorrow).

In contrast to these works, while buyers in our model do care about the future (e.g. they value learning), they don't reason about how their actions today might affect the seller's decisions tomorrow. Our model better captures settings where full information about the auction is not public (and fully strategic reasoning is simply impossible without the necessary information).

Other related work considers the *Price of Anarchy* of simple combinatorial auctions when bidders no-regret learn [173, 181, 160, 68]. One key difference between this line of work and ours is that these all study welfare maximization for combinatorial auctions with rich valuation functions. In contrast, our work studies revenue maximization while selling a single item. Additionally, in these works the seller commits to a publicly known auction format, and the only reason for learning is due to the strategic behavior of other buyers. In contrast, buyers in our model have to learn *even when they are the only buyer*, due to the strategic nature of the seller.

Recent work has also considered learning from the perspective of the seller. In these works, the buyer's (or buyers') valuations are drawn from an unknown distribution, and the seller's goal is to learn an approximately optimal auction with as few samples as possible [65, 71, 155, 156, 109, 47, 85]. These works consider numerous different models and achieve a wide range of guarantees, but all study the learning problem from the perspective of the *seller*, whereas the buyer is simply myopic and participates in only one round. In contrast, it is the buyer in our model who does the learning (and there is no information for the seller to learn: the buyer's values are drawn fresh in every round).

Finally, no-regret learning in online decision problems is an extremely well-studied problem. When feedback is revealed for every possible action, one well-known solution is the multiplicative weight update rule which has been rediscovered and applied in many fields (see survey [14] for more details). Another algorithmic scheme for the online decision problem is known as Follow the Perturbed Leader [114, 128, 129]. When only feedback for the selected action is revealed, the problem is referred to as the multi-armed bandit problem. Here, similar ideas to the MWU rule are used in developing the EXP3 algorithm [18] for adversarial bandit model, and also for the contextual bandit problem [138]. Our algorithm in Theorem 2.3.2 bears some similarities to the low swap regret algorithm introduced in [28].

See the survey [45] for more details about the multi-armed bandit problem. Our results hold in both models (i.e. whether the buyer receives feedback for every bid they could have made, or only the bid they actually make), so we will make use of both classes of algorithms.

In summary, while there is already extensive work related to repeated sales in auctions, and even no-regret learning with respect to auctions (from both the buyer and seller perspective), our work is the first to address how a seller might adapt their selling strategy when faced with a no-regret buyer.

2.2 Model and Preliminaries

We consider a setting with 1 buyer and 1 seller. There are T rounds, and in each round the seller has one item for sale. At the start of each round t, the buyer's value $v(t)$ (known only to the buyer) for the item is drawn independently from some distribution $\mathcal{D}$ (known to both the seller and the buyer). For simplicity, we assume $\mathcal{D}$ has a finite support[3] of size m, supported on values $0 \leq v_1 < v_2 < \cdots < v_m \leq 1$. For each $i \in [m]$, v_i has probability q_i of being drawn under $\mathcal{D}$.

The seller then presents K options for the buyer, which can be thought of as "possible bids" (we will interchangeably refer to these as *options*, *bids*, or *arms* throughout the paper, depending on context). Each arm i is labelled with a bid value $b_i \in [0, 1]$, with $b_1 < \ldots, < b_K$. Upon pulling this arm at round t, the buyer receives the item with some allocation probability $a_{i,t}$, and must pay a price $p_{i,t} \in [0, a_{i,t} \cdot b_i]$. These values $a_{i,t}$ and $p_{i,t}$ are chosen by the seller during time t, but remain unknown to the buyer until he plays an arm, upon which he learns the values for that arm. All of our positive results (i.e. strategies for the seller) are *non-adaptive* (in some places called *oblivious*), in the sense that that $a_{i,t}, p_{i,t}$ are set before the first round starts. All of our negative results (i.e. upper bounds on how much a seller

[3] If $\mathcal{D}$ instead has infinite support, all our results hold approximately after discretization to multiples of ε. If $\mathcal{D}$ is bounded in $[0, H]$, then all our results hold after normalizing $\mathcal{D}$ by dividing by H.

can possibly attain) hold even against *fully adaptive* sellers, where $a_{i,t}$ and $p_{i,t}$ can be set *even after learning the distribution of arms the buyer intends to pull in round t.*

In order for the selling strategies to possibly represent natural auctions, we require the allocation/price rules to be monotone. That is, if $i > j$, then for all t, $a_{i,t} \geq a_{j,t}$ and $p_{i,t} \geq p_{j,t}$. In other words, bidding higher should result in a (weakly) higher probability of receiving the item and (weakly) higher expected payment. We'll also insist on the existence of an arm 0 with bid $b_0 = 0$ and $a_{0,t} = 0$ for all t; i.e., an arm which charges nothing but does not give the item. Playing this arm can be thought of as not participating in the auction.

2.2.1 Bandits and experts

Our goal is to understand the behavior of such mechanisms when the buyer plays according to some no-regret strategy for the multi-armed bandit problem. In the classic multi-armed bandit problem a learner (in our case, the buyer) chooses one of K arms per round, over T rounds. On round t, the learner receives a reward $r_{i,t} \in [0, 1]$ for pulling arm i (where the values $r_{i,t}$ are possibly chosen adversarially). The learner's goal is to maximize his total reward.

Let I_t denote the arm pulled by the principal at round t. The *regret* of an algorithm $\mathcal{A}$ for the learner is the random variable $\mathsf{Reg}(\mathcal{A}) = \max_i \sum_{t=1}^{T} r_{i,t} - \sum_{t=1}^{T} r_{I_t,t}$. We say an algorithm $\mathcal{A}$ for the multi-armed bandit problem is *δ-no-regret* if $\mathrm{E}[\mathsf{Reg}(\mathcal{A})] \leq \delta$ (where the expectation is taken over the randomness of $\mathcal{A}$). We say an algorithm $\mathcal{A}$ is *no-regret* if it is δ-no-regret for some $\delta = o(T)$.

In the multi-armed bandits setting, the learner only learns the value $r_{i,t}$ for the arm i which he pulls on round t. In our setting, the learner will learn $a_{i,t}$ and $p_{i,t}$ explicitly (from which they can compute $r_{i,t}$). Our results (both positive and negative) also hold when the learner learns the value $r_{i,t}$ for *all* arms i (we refer this full-information setting as the *experts setting*, in contrast to the partial-information *bandits setting*). Simple no-regret algorithms exist in both the experts setting and the bandits setting. Of special interest in this paper will be a class of learning algorithms for the bandits problem and experts problem which we term 'mean-based'.

Definition 2.2.1 (Mean-Based Learning Algorithm). *Let $\sigma_{i,t} = \sum_{s=1}^{t} r_{i,s}$. An algorithm for the experts problem or multi-armed bandits problem is γ-mean-based if it is the case that whenever $\sigma_{i,t} < \sigma_{j,t} - \gamma T$, then the probability that the algorithm pulls arm i on round t is at most γ. We say an algorithm is mean-based if it is γ-mean-based for some $\gamma = o(1)$.*

Intuitively, 'mean-based' algorithms will rarely pick an arm whose current mean is significantly worse than the current best mean. Many no-regret algorithms, including commonly used variants of EXP3 (for the bandits setting), the Multiplicative Weights algorithm (for the experts setting) and the Follow-the-Perturbed-Leader algorithm (experts setting), are mean-based (see the full version of [37]).

Contextual bandits

In our setting, the buyer has the additional information of their current value for the item, and hence is actually facing a *contextual bandits* problem. In (our variant of) the contextual bandits problem, each round t the learner is additionally provided with a *context* c_t drawn from some distribution $\mathcal{D}$ supported on a finite set C (in our setting, $c_t = v(t)$, the buyer's valuation for the item at time t). The adversary now specifies rewards $r_{i,t}(c)$, the reward the learner receives if he pulls arm i on round t while having context c. If we are in the full-information (experts) setting, the learner learns the values of $r_{i,t}(c_t)$ for all arms i after round t, where as if we are in the partial-information (bandits) setting, the learner only learns the value of $r_{i,t}(c_t)$ for the arm i that he pulled.

In the contextual bandits setting, we now define the regret of an algorithm $\mathcal{A}$ in terms of regret against the best "context-specific" policy π; that is, $\mathsf{Reg}(\mathcal{A}) = \max_{\pi:C\to[K]} \sum_{t=1}^{T} r_{\pi(c_t),t}(c_t) - \sum_{t=1}^{T} r_{I_t,t}(c_t)$, where again I_t is the arm pulled by M on round t. As before, we say an algorithm is δ-low regret if $\mathrm{E}[\mathsf{Reg}(M)] \leq \delta$, and say an algorithm is no-regret if it is δ-no-regret for some $\delta = o(T)$.

If the size of the context set C is constant with respect to T, then there is a simple way to construct a no-regret algorithm M' for the contextual bandits problem from a no-regret algorithm M for the classic bandits problem: simply maintain a separate instance of M for every different context $v \in C$ (in the contextual bandits literature, this is sometimes referred to as the S-EXP3 algorithm [45]). We call the algorithm we obtain this way its *contextualization*, and denote it as $\mathsf{cont}(M)$.

If we start with a mean-based learning algorithm, then we can show that its contextualization satisfies an analogue of the mean-based property for the contextual-bandits problem.

Definition 2.2.2 (Mean-Based Contextual Learning Algorithm). *Let $\sigma_{i,t}(c) = \sum_{s=1}^{t} r_{i,s}(c)$. An algorithm for the contextual bandits problem is γ-mean-based if it is the case that whenever $\sigma_{i,t}(c) < \sigma_{j,t}(c) - \gamma T$, then the probability $p_{i,t}(c)$ that the algorithm pulls arm i on round t if it has context c satisfying $p_{i,t}(c) < \gamma$. We say an algorithm is mean-based if it is γ-mean-based for some $\gamma = o(1)$.*

Theorem 2.2.1. *If an algorithm for the experts problem or multi-armed bandits problem is mean-based, then its contextualization is also a mean-based algorithm for the contextual bandits problem.*

Finally, we will refer to learning algorithms that never overbid as *conservative*. We will sometimes abuse notation and instead refer to a buyer employing a conservative algorithm as conservative.

2.2.2 Welfare and monopoly revenue

In order to evaluate the performance of our mechanisms for the seller, we will compare the revenue the seller obtains to two benchmarks from the single-round setting of a seller selling a single item to a buyer with value drawn from distribution $\mathcal{D}$.

The first benchmark we consider is the *welfare* of the buyer, the expected value the buyer assigns to the item. This quantity clearly upper bounds the expected revenue that the seller can hope to extract per round.

Definition 2.2.3. *The* welfare, $\mathsf{Val}(\mathcal{D})$ *is equal to* $\mathbb{E}_{v\sim\mathcal{D}}[v]$.

The second benchmark we consider is the *monopoly revenue*, the maximum possible revenue attainable by the seller in one round against a rational buyer. Seminal work of Myerson [157] shows that this revenue is attainable by setting a fixed price ("monopoly/Myerson reserve") for the item, and hence can be characterized as follows.

Definition 2.2.4. *The* monopoly revenue *(alternatively,* Myerson revenue*)* $\mathsf{Mye}(\mathcal{D})$ *is equal to* $\max_p p \cdot \Pr_{v\sim\mathcal{D}}[v \geq p]$.

2.2.3 A final note on the model

For concreteness, we chose to phrase our problem as one where a single bidder whose value is repeatedly drawn independently from $\mathcal{D}$ each round engages in no-regret learning with their value as context. Alternatively, we could imagine a population of m different buyers, each with a *fixed* value v_i. Each round, exactly one buyer arrives at the auction, and it is buyer i with probability q_i. The buyers are indistinguishable to the seller, and each buyer no-regret learns (without context, because their value is always v_i). This model is mathematically equivalent to ours, so all of our results hold in this model as well if the reader prefers this interpretation instead.

2.3 An Illustrative Example

In this section, we overview an illustrative example to show the difference between mean-based and non-mean-based learning algorithms, and between conservative and non-conservative learners. We will not prove all claims in this section (nor carry out all calculations) as it is only meant to illustrate and provide intuition. Throughout this section, the running example will be when $\mathcal{D}$ samples 1/4 with probability 1/2, 1/2 with probability 1/4, and 1 with probability 1/4. Note that $\mathsf{Val}(\mathcal{D}) = 1/2$ and $\mathsf{Rev}(\mathcal{D}) = 1/4$.

2.3.1 Mean-Based Learning

Let's first consider what the seller can do with an auction when the buyer is running a mean-based (non-conservative) learning algorithm like EXP3. The seller will let the buyer bid 0 or 1. If the buyer bids 0, they pay nothing but do not receive the item (recall that an arm of this form is required). If the buyer bids 1 in round t, they receive the item and pay some price p_t as follows: for the first half of the game ($1 \leq t \leq T/2$), the seller sets $p_t = 0$. For the second half of the game ($T/2 < t \leq T$), the seller sets $p_t = 1$.

Let's examine the behaviour of the buyer, recalling that they run a mean-based learning algorithm, and therefore (almost) always pull the arm with highest cumulative utility. The buyer with value 1 will happily bid 1 all the way through, since he is always offered the item for less than or equal to his value for the item. The buyer with value 1/2 will bid 1 for the first $T/2$ rounds, accumulating a surplus (i.e., negative regret) of 1/2 per round. For the next $T/2$ rounds, this surplus slowly disappears at the rate of 1/2 per round until it disappears at time T, so the bidder with value 1/2 will bid 1 all the way through. Finally, the bidder with value 1/4 will bid 1 for the first $T/2$ rounds, accumulating surplus at a rate of 1/4 per round. After round $T/2$, this surplus decreases at a rate of 3/4 per round, until at round $2T/3$ his cumulative utility from bidding 1 reaches 0 and he switches to bidding 0.

Now let's compute the revenue. From round $T/2$ through $2T/3$, the buyer always buys the item at a price of 1, so the seller obtains $T/6$ revenue. Finally, from round $2T/3$ through T, the buyer purchases the item with probability 1/2 and pays 1. The total revenue is $0 + T/6 + T/6 = T/3$. Note that if the seller used the default strategy, they would extract revenue only $T/4$.

Where did our extra revenue come from? First, note that the welfare of the buyer in this example is quite high: the bidder gets the item the whole way through when $v \geq 1/2$, and two-thirds of the way through when $v = 1/4$. One reason why the welfare is so high is because we give the item away for free in the early rounds. But notice also that the utility of the buyer is quite low: the buyer actually has zero utility when $v \leq 1/2$, and utility 1/2 when $v = 1$. The reason we're able to keep the utility low, despite giving the item away for free in the early rounds is because we overcharge the bidders in later rounds (and they choose to overpay, exactly because their learning is mean-based).

In fact, by offering additional options to the buyer, we show that *it is possible for the seller to extract up to the full welfare from the buyer* (e.g. a net revenue of $T/2 - o(T)$ for this example). As in the above example, our mechanism makes use of arms which are initially very good for the buyer (giving the item away for free, accumulating negative regret), followed by a period where they are very bad for the buyer (where they pay more than their value). The trick in the construction is making sure that the good/bad intervals line up so that: a) the buyer purchases the item in every round, no matter their value (this is necessary in order to possibly extract full welfare) and b) by round T, the buyer has zero (arbitrarily small) utility, no matter their value.

Getting the intervals to line up properly so that any mean-based learner will pick the desired arms still requires some work. But interestingly, our constructed mechanism is non-adaptive and prior-independent (i.e. the same mechanism extracts full welfare *for all* $\mathcal{D}$). Theorem 2.3.1 below formally states the guarantees. The construction itself and the proof appear in the full version of [37].

Theorem 2.3.1. *If the buyer is running a mean-based algorithm, for any constant* $\varepsilon > 0$*, there exists a strategy for the seller which obtains revenue at least* $(1-\varepsilon)\mathsf{Val}(\mathcal{D})T - o(T)$*.*

Two properties should jump out as key in enabling the result above. The first is that the buyer *only* has no regret towards fixed arms and *not* towards the policy they would have used with a lower value (this is what leads the buyer to continue bidding 1 with value 1/2 even though they have already learned to bid 0 with value 1/4). This suggests an avenue towards an improved learning algorithm: have the bidder attempt to have no regret not only towards each fixed arm, but also towards the policy of play produced when having different values. This turns out to be exactly the right idea, and is discussed in the following subsection below.

The second key property is that we were able to "lure" the bidders into playing an arm with a free item, then overcharge them later to make up for lost revenue. This requires that the bidder consider pulling an arm with maximum bid exceeding their value, which will never happen for a conservative bidder. It turns out it is still possible to do better than the default strategy against conservative bidders, but not as well as against non-conservative mean-based bidders. Section 2.3.3 explores conservative mean-based bidders for this example.

2.3.2 Better Learning

In our bad example above, the buyer with value 1/2 for the item slowly spends the second half of the game losing utility. While his behaviour is still no-regret (he ends up with zero net utility, which indeed is at least as good as only bidding 0), he would have been much happier to follow the actions of the buyer with value 1/4, who started bidding 0 at $2T/3$.

Using this idea, we show how to construct a no-regret algorithm for the buyer (Algorithm 1) such that the seller receives at most the Myerson revenue every round. We accomplish this by extending an arbitrary no-regret algorithm (e.g. EXP3) by introducing "virtual arms" for each value, so that each buyer with value v has low regret not just with respect to every fixed bid, but also no-regret with respect to the policy of play as if they had a different value v' for the item (for all $v' < v$). In some ways, our construction is very similar to the construction of low internal-regret (or swap-regret) algorithms from low external-regret algorithms. The main difference is that instead of having low regret with respect to swapping actions, we have low regret with respect to swapping *contexts* (i.e. values). Theorem 2.3.2 below states that the seller cannot outperform the default strategy against buyers who use such algorithms to learn.

Theorem 2.3.2. *There exists a no-regret algorithm (Algorithm 1) for the buyer against which every seller strategy extracts no more than* $\mathsf{Mye}(\mathcal{D})T + O(m\sqrt{\delta T})$ *revenue.*

Algorithm 1 No-regret algorithm for buyer where the seller achieves no more than $\mathsf{Mye}(\mathcal{D})T + o(T)$ revenue.

Let M be a δ-no-regret algorithm for the classic multi-armed bandit problem, with $\delta = o(T)$. Initialize m copies of M, M_1 through M_m.
Instance M_i of M will learn over $K + i - 1$ arms.
The first K arms of M_i ("bid arms") correspond to the K possible menu options $b_1, b_2, \ldots, b_K$.
The last $i-1$ arms of M_i ("value arms") correspond to the $i-1$ possible values (contexts) $v_1, \ldots, v_{i-1}$.
for $t = 1$ to T **do**
  if buyer has value v_i **then**
    Use M_i to pick one arm from the $K + i - 1$ arms.
    if the arm is a bid arm b_j **then**
      Pick the menu option j (i.e. bid b_j).
    else if the arm is a value arm v_j **then**
      Sample an arm from M_j (but don't update its state). If it is a bid arm, pick the corresponding menu option. If it is a value arm, recurse.
    end if
    Update the state of algorithm M_i with the utility of this round.
  end if
end for

A more further discussion of the algorithm along with a proof of Theorem 2.3.2 appear in the full version of [37]. The key observation in the proof is that "not regretting playing as if my value were v'" sounds a lot like "not preferring to report value v' instead of v." This suggests that the aggregate allocation probabilities and prices paid by any buyer using our algorithm should satisfy the same constraints as a truthful auction, proving that the resulting revenue cannot exceed the default strategy (and indeed the proof follows this approach).

Finally, observe that the following corollary immediately follows. Because the seller cannot hope to get more than $\mathsf{Mye}(\mathcal{D})T+o(T)$ per round when the buyer is using Algorithm 1, and the buyer cannot hope to do better than telling the truth against a truthful auction, it is in fact a Nash for the buyer to use Algorithm 1 and the seller to set price equal to the Myerson reserve every round.

Corollary 2.3.1. *It is an $o(T)$-Nash equilibrium for the seller to set the Myerson reserve $p(\mathcal{D})$ in every round (any bid $\geq p(\mathcal{D})$ reserve wins the item and pays $p(\mathcal{D})$), and the buyer to use Algorithm 1.*

2.3.3 Mean-Based Learning and Conservative Bidders

Recall in our example that to extract revenue $T/3$, bidders with values $1/4$ and $1/2$ had to consider bidding 1. If bidders are conservative, they will simply never do this.

Although the auction in Section 2.3.1 is no longer viable, consider the following auction instead: in addition to the zero arm, the bidder can bid $1/4$ or $1/2$. If they bid $1/2$ in any round, they will get the item with probability 1 and pay $1/2$. If they bid $1/4$ in round $t \leq T/3$, they get nothing. If they bid $1/4$ in round $t \in (T/3, T]$, they get the item and pay $1/4$. Let's again see what the bidder will choose to do, remembering that they will always pull the arm that has provided highest cumulative utility (due to being mean-based).

Clearly, the bidder with value $1/4$ will bid $1/4$ every round (since they are conservative, they won't even consider bidding $1/2$), making a total payment of $2T/3 \cdot 1/4 \cdot 1/2 = T/12$. The bidder with value $1/2$ will bid $1/2$ for the first $T/3$ rounds, and then immediately switch to bidding $1/4$, making a total payment of $T/3 \cdot 1/2 \cdot 1/4 + 2T/3 \cdot 1/4 \cdot 1/4 = T/12$.

The bidder with value 1 will actually bid $1/2$ for the entire T rounds. To see this, observe that their cumulative surplus through round t from bidding $1/2$ is $t \cdot 1/2 \cdot 1/4 = t/8$ (t rounds by utility $1/2$ per round by probability $1/4$ of having value 1). Their cumulative surplus through round t from bidding $1/4$ is instead $(t - T/3) \cdot 3/4 \cdot 1/4 = 3t/16 - T/16 \leq t/8$ (for $t \leq T$). Because they are mean-based, they will indeed bid $1/2$ for the entire duration due to its strictly higher utility. So their total payment will be $T \cdot 1/2 \cdot 1/4 = T/8$. The total revenue is then $7T/24 > T/4$, again surpassing the default strategy (but not reaching the $T/3$ achieved against non-conservative buyers).

Let's again see where our extra revenue comes from in comparison to a truthful auction. Notice that the bidder receives the item with probability 1 conditioned on having value 1/2, and also conditioned on having value 1. Yet somehow the bidder pays an average of 1/3 conditioned on having value 1/2, but an average of 1/2 conditioned on having value 1. *This could never happen in a truthful auction*, as the bidder would strictly prefer to pretend their value was 1/2 rather than 1. But it is entirely possible when the buyer does mean-based learning, as evidenced by this example.

We define $\mathsf{MBRev}(\mathcal{D})$ as the value of the LP in Figure 2.1. In Theorems 2.3.4 and 2.3.3, we show that $\mathsf{MBRev}(\mathcal{D})T$ tightly characterizes (up to $\pm o(T)$) the optimal revenue a seller can extract against a conservative buyer. The proofs can be found in the full version of [37].

$$\begin{aligned} \textbf{maximize} \quad & \sum_{i=1}^{m} q_i(v_i x_i - u_i) \\ \textbf{subject to} \quad & u_i \geq (v_i - v_j) \cdot x_j, \quad \forall\, i, j \in [m] : i > j \\ & u_i \geq 0, 1 \geq x_i \geq 0, \quad \forall\, i \in [m] \end{aligned}$$

Figure 2.1: The mean-based revenue LP.

Before stating our theorems, let us parse this LP. q_i is a constant representing the probability that the buyer has value v_i (also a constant). x_i is a variable representing the average probability that the bidder gets the item with value v_i, and u_i is a variable representing the average utility of the bidder when having value v_i. Therefore, this bidder's average value is $v_i x_i$, the average price they pay is $v_i x_i - u_i$, and the objective function is simply the average revenue. The second constraints are just normalization, ensuring that everything lies in $[0, 1]$. The first line of constraints are the interesting ones. These look a lot like IC constraints that a truthful auction must satisfy, but something's missing: the LHS is clearly the utility of the buyer with value v_i for "telling the truth," but the utility of the buyer for "reporting v_j instead" is $(v_i - v_j) \cdot x_j + u_j$ (so the u_j term is missing on the RHS).

Here is a brief proof outline for why no seller can extract more revenue than $\mathsf{MBRev}(\mathcal{D})$:

1. Since the buyer has no regret conditioned on having value v_i, their utility is at least as high as playing arm j every round, for all $j \leq i$.

2. Since the auction never charges arm j more than v_j (conditioned on awarding the item), the buyer's utility for playing arm j every round is at least $y_j \cdot (v_i - v_j)$, where y_j is the average probability that arm j awards the item.

3. Since the auction is monotone, and the buyer never considers overbidding, if the buyer gets the item with probability x_j conditioned on having value v_j, we must have $y_j \geq x_j$.

These three facts together show that no seller can extract more than $\mathsf{MBRev}(\mathcal{D})$ against a no-regret buyer who doesn't overbid. Observe also that step 3 is *exactly* the step that doesn't hold for buyers who consider overbidding (and is exactly what's violated in our example in Section 2.3.1): if the buyer ever overbids, then they might receive the item with higher probability than had they just played their own arm every round.

Theorem 2.3.3. *Any strategy for the seller achieves revenue at most* $\mathsf{MBRev}(\mathcal{D})T + o(T)$ *against a conservative buyer.*

The full proof of Theorem 2.3.3 can be found in the full version of [37] - all of the key ideas have been overviewed above.

It turns out that the previous theorem is tight; there exists an auction (taking the form of a first-price auction with descending reserve) which achieves revenue $\mathsf{MBRev}(\mathcal{D})T$ against a conservative mean-based buyer. More specifically, this auction is defined by a threshold r_t that decreases over time. If at time t you bid $b_t \geq r_t$, then you receive the item and must pay b_t; otherwise, you receive nothing and pay nothing. Moreover, the threshold function r_t which achieves optimal revenue is determined from the optimal solution to the mean-based LP: the threshold r_t drops from v_i to v_{i+1} at round x_i (where the x_i belong to some optimal solution).

To show that this is a valid strategy for the seller, we need to show that the values x_i are monotone increasing. Luckily, this follows simply from the structure of the mean-based revenue LP.

Lemma 2.3.1. *Let $x_1, x_2, \ldots, x_m, u_1, u_2, \ldots, u_m$ be an optimal solution to the mean-based revenue LP. Then for all $i < j$, $x_i < x_j$.*

Proof. We proceed by contradiction. Suppose that the sequence of x_i are not monotone; then there exists an $1 \leq i \leq m-1$ such that $x_i > x_{i+1}$. Now consider another solution of the LP, where we increase x_{i+1} to x_i, keeping the value of all other variables the same. This new solution does not violate any constraints in the LP since for all $j > i+1$, $u_j \geq (v_j - v_i) \cdot x_i \geq (v_j - v_{i+1}) \cdot x_i$. However this change increases the value of the objective by $v_{i+1}q_{i+1}(x_i - x_{i+1}) > 0$, thus contradicting the fact that $x_1, \ldots, x_m, u_1, ..., u_m$ was an optimal solution of the mean-based revenue LP. □

We now show that this strategy indeed achieves $\mathsf{MBRev}(\mathcal{D})T$ against a conservative buyer.

Theorem 2.3.4. *For any constant $\varepsilon > 0$, there exists a strategy for the seller gets revenue at least $(\mathsf{MBRev}(\mathcal{D}) - \varepsilon)T - o(T)$ against a buyer running a mean-based algorithm who overbids with probability 0. The strategy sets a decreasing cutoff r_t and for all t awards the item with probability 1 to any bid $b_t \geq r_t$ for price b_t, and with probability 0 to any bid $b_t < r_t$.*

Proof. We will show that: i) the buyer with value v_i receives the item for at least $x_iT - o(T)$ turns (receiving $v_ix_iT - o(T)$ total utility from the items), and ii) this buyer's net utility is at most $(u_i + \varepsilon)T + o(T)$. This implies that this buyer pays the seller at least $x_iv_iT - (u_i + \varepsilon)T - o(T)$ over the course of the T rounds; taking expectation over all v_i completes the proof.

Assume the buyer is running a γ-mean-based learning algorithm. Consider the buyer when they have value v_i. Note that

$$\sigma_{j,t}(v_i) = (v_i - v_j + \varepsilon) \cdot \max(0, t - (1 - x_j)T).$$

We first claim that after round $(1 - x_i)T + \gamma T/\varepsilon$, the buyer will buy the item (i.e., choose an option that results in him getting the item) each round with probability at least $1 - m\gamma$. To see this, first note that $\sigma_{i,t}(v_i) \geq \gamma T$ when $t \geq (1 - x_i)T + \gamma T/\varepsilon$. Then, since the cumulative utility of any arm is 0 until it starts offering the item, it follows from the mean-based condition that the buyer will pick a specific arm that is not offering the item with probability at most γ, and therefore choose some good arm with probability at least $1-m\gamma$. It follows that, in expectation, the buyer with value v_i receives the item for at least $(1 - m\gamma)(x_iT - \gamma T/\varepsilon) = x_iT - o(T)$ turns.

We now proceed to upper bound the overall expected utility of the buyer. For each index $j \leq i$, let S_j be the set of t where $\sigma_{j,t}(v_i) > \sigma_{j',t}(v_i)$ for all other j'. Note that since each $\sigma_{j,t}(v_i)$ is a linear function in t (when positive), each S_j is either the empty set or an interval (y_jT, z_jT). Since all the v_i are distinct, note that these intervals partition the interval $((1 - x_i)T, T)$ (with the exception of up to m endpoints of these intervals); in particular, $\sum_{j\geq i}(z_j - y_j) = x_i$.

Let $\varepsilon' = \min_j(v_{j+1} - v_j)$. Note that, if $t \in (y_jT + \gamma T/\varepsilon', z_jT - \gamma T/\varepsilon')$, then for all $j' \neq j$, $\sigma_{j,t}(v_i) > \sigma_{j',t}(v_i) + \gamma T$. This follows since $\sigma_{j,t}(v_i) - \sigma_{j',t}(v_i)$ is linear in t with slope $v_j - v_{j'}$, and $|v_j - v_{j'}| > \varepsilon'$. It follows that if t is in this interval, then the buyer will choose option j with probability at least $1 - m\gamma$ (by a similar argument as before).

Define $j(t) = \arg\max_j \sigma_{j,t}(v_i)$ to be the index of the arm with the current largest cumulative reward, and let $\sigma_{max,t}(v_i) = \sum_{s=1}^{t} r_{j(s),s}(v_i)$ be the cumulative utility of always playing the arm with the current highest cumulative reward for the first t rounds. The following lemma shows that $\sigma_{max,T}(v_i)$ is close to $\max_j \sigma_{j,T}(v_i)$. (In other words, playing the best arm every round and playing the best-at-the-end arm every round have similar payoffs if the historically best arm does not change often).

Lemma 2.3.2. *$|\sigma_{max,T}(v_i) - \max_j \sigma_{j,T}(v_i)| \leq m$.*

Proof. Let $W = |\{t | j(t) \neq j(t+1)\}|$ equal the number of times the best arm switches values; note that since each $\sigma_{j,t}(v_i)$ is linear, W is at most m. Let $t_1 < t_2 < \cdots < t_W$ be the values of t such that $j(t) \neq j(t+1)$. Additionally define $t_0 = 1$ and $t_{W+1} = T$. Then, dividing the cumulative reward $\sigma_{max,t}$ into intervals by these t_i, we get that

$$\begin{aligned}
\sigma_{max,t}(v_i) &= \sum_{s=1}^{t} r_{j(s),s}(v_i) \\
&= \sum_{i=1}^{W+1} (\sigma_{j(t_i),t_i}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i)) \\
&= \sigma_{j(T),T}(v_i) + \sum_{i=1}^{W+1} (\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i)) \\
&= \max_j \sigma_{j,t}(v_i) + \sum_{i=1}^{W+1} (\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i))
\end{aligned}$$

It therefore suffices to show that $|\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i)| \leq 1$ for all i. To see this, note that (by the definition of $j(t)$), $\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i) > 0$, and that $\sigma_{j(t_{i-1}),t_{i-1}+1}(v_i) - \sigma_{j(t_i),t_{i-1}+1}(v_i) < 0$. However,

$$\begin{aligned}
&(\sigma_{j(t_{i-1}),t_{i-1}+1}(v_i) - \sigma_{j(t_i),t_{i-1}+1}(v_i)) \\
=&(\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i)) + (r_{j(t_{i-1}),t_{i-1}+1}(v_i) - r_{j(t_i),t_{i-1}+1}(v_i))
\end{aligned}$$

Since $0 \leq r_{j,t}(u) \leq 1$, it follows that $|\sigma_{j(t_{i-1}),t_{i-1}}(v_i) - \sigma_{j(t_i),t_{i-1}}(v_i)| \leq 1$. This completes the proof. □

Let $\sigma_T(v_i) = \sum_{t=1}^{T} \mathrm{E}[r_{I_t,t}(v_i)]$ denote the expected cumulative utility of this buyer at time T. We claim that $\sigma_T \leq \max_j \sigma_{j,T}(v_i) + o(T)$. To see this, recall that, for $t \in (y_j T + \gamma T/\varepsilon', z_j T - \gamma T/\varepsilon')$, $\Pr[I_t \neq j] \leq m\gamma$, and therefore $\mathrm{E}[r_{I_t,t}] \leq r_{j,t} + m\gamma$. Furthermore, note that for $t \in S_j$, $j(t) = j$, so $r_{j,t} = r_{j(t),t}$ and $\mathrm{E}[r_{I_t,t}] \leq r_{j(t),t} + m\gamma$. It follows that

$$
\begin{aligned}
\sigma_T(v_i) &= \sum_{t=1}^{T} \mathrm{E}[r_{I_t,t}(v_i)] \\
&\le \sum_{t=(1-x_i)T}^{T} \mathrm{E}[r_{I_t,t}(v_i)] \\
&= \sum_{j=1}^{i} \sum_{t=y_jT}^{z_jT} \mathrm{E}[r_{I_t,t}(v_i)] \\
&\le \sum_{j=1}^{i} \left(\frac{2\gamma T}{\varepsilon'} + \sum_{t=y_jT+\gamma T/\varepsilon'}^{z_jT-\gamma T/\varepsilon'} \mathrm{E}[r_{I_t,t}(v_i)] \right) \\
&\le \sum_{j=1}^{i} \left(\frac{2\gamma T}{\varepsilon'} + \sum_{t=y_jT+\gamma T/\varepsilon'}^{z_jT-\gamma T/\varepsilon'} (r_{j(t),t}(v_i) + m\gamma) \right) \\
&\le \frac{2m\gamma T}{\varepsilon'} + m\gamma T + \sum_{t=1}^{T} r_{j(t),t}(v_i) \\
&= \frac{2m\gamma T}{\varepsilon'} + m\gamma T + \sigma_{max,T}(v_i) \\
&\le \frac{2m\gamma T}{\varepsilon'} + m\gamma T + m + \max_j \sigma_{j,T}(v_i) \\
&= \max_j \sigma_{j,T}(v_i) + o(T).
\end{aligned}
$$

Finally, note that

$$
\begin{aligned}
\max_j \sigma_{j,T}(v_i) &= \max_{j<i} (v_i - v_j + \varepsilon) x_j T \\
&\le (\max_{j<i} (v_i - v_j) x_j + \varepsilon) T \\
&= (u_i + \varepsilon) T
\end{aligned}
$$

It follows that $\sigma_T(v_i) \le (u_i + \varepsilon)T + o(T)$, as desired.

□

Finally, we show that this quantity $\mathsf{MBRev}(\mathcal{D})$ is in fact significantly different from both $\mathsf{Val}(\mathcal{D})$ and $\mathsf{Rev}(\mathcal{D})$; in particular, it is a constant-factor approximation to neither. In particular, the multiplicative gap between $\mathsf{MBRev}(\mathcal{D})$ and $\mathsf{Rev}(\mathcal{D})$ can grow as large as $\log\log H$ for distributions $\mathcal{D}$ supported on $[1, H]$. In comparison, the gap between $\mathsf{Val}(\mathcal{D})$ and $\mathsf{Rev}(\mathcal{D})$ can grow as large as $\log H$ on this same interval, and in fact both gaps are maximized for the same distribution: the equal-revenue curve $\mathcal{D}_{ERC}$ truncated at H.

Theorem 2.3.5. *For distributions $\mathcal{D}$ supported on $[1, H]$, $\mathsf{MBRev}(\mathcal{D}) = O(\log\log H)$, and there exist $\mathcal{D}$ supported on $[1, H]$ such that $\mathsf{MBRev}(\mathcal{D}) = \Theta(\log\log H)$. For this same $\mathcal{D}$, $\mathsf{Val}(\mathcal{D}) = \Theta(\log H)$.*

The proof of Theorem 2.3.5 can be found in the full version of [37]. The proof is divided into two parts (after extending the definition of $\mathsf{MBRev}(\mathcal{D})$ to hold for continuous distributions $\mathcal{D}$): 1. showing that $\mathsf{MBRev}(\mathcal{D}_{ERC}) \leq O(\log\log H)$, and 2. showing that $\mathsf{MBRev}(\mathcal{D}_{ERC}) \geq O(\log\log H)$.

To show the first part, it suffices to simply demonstrate a solution to the mean-based LP with value at least $O(\log\log H)$. In the proof it suffices to choose $x(v) = \frac{\log v}{\log H}$ (equivalently, the reserve for the associated second-price auction should exponentially decay over time).

To show the second part, we examine the dual of the LP. Effectively, this involves rewriting $\mathsf{MBRev}(\mathcal{D})$ in the form

$$\mathsf{MBRev}(\mathcal{D}) = \max_x \mathbb{E}_{v_i \sim \mathcal{D}} \left[v_i x_i - \max_j (v_i - v_j) x_j \right]$$

(in particular, note that for a fixed choice of x, $u_j = \max_j (v_i - v_j) x_j$), and finding an appropriate function $j(i)$ (which corresponds to an assignment to the dual).

2.3.4 A Final Note on the Example

While reading through our examples, the reader may think that the mean-based learner's behavior is clearly irrational: why would you continue paying above your value? Why would you continue paying more than necessary, when you can safely get the item for less?

But this is exactly the point: a more thoughtful learner can indeed do better (for instance, by using the algorithm of Section 2.3.2). It is also perhaps misleading to believe that the bidder should "obviously" stop overpaying: we only know this because we know the structure of the example. But in principle, how is the bidder supposed to know that the overcharged rounds are the new norm and not an anomaly? Given that most standard no-regret algorithms are mean-based, it's important to nail down the seller's options for exploiting this behavior.

2.4 Conclusion and Future Directions

We consider a revenue-maximizing seller with a single item (each round) to sell to a single buyer. We show that when the buyer uses mean-based algorithms like EXP3, the seller can extract revenue equal to the expected welfare with an unnatural auction. We then provide a modified no-regret algorithm $\mathcal{A}$ such that the seller cannot extract revenue exceeding the monopoly revenue when the buyer bids according to $\mathcal{A}$. Finally, we consider a mean-based buyer who never overbids. We tightly characterize the seller's optimal revenue with a linear program, and show that a pay-your-bid auction with decreasing reserves over time achieves this guarantee. Moreover, we show that the mean-based revenue can be unboundedly better than the monopoly revenue while simultaneously worse than the expected welfare. In particular, for the equal revenue curve truncated at H, the monopoly revenue is 1, the expected welfare is $\ln(H)$, and the mean-based revenue is $\Theta(\ln(\ln(H)))$.

While our work has already shown the single-buyer problem is quite interesting, the most natural direction for future work is understanding revenue maximization with multiple learning buyers. Of our three main results, only Theorem 2.3.2 extends easily (that if every buyer uses our modified learning, the default strategy, which now runs Myerson's optimal auction every round, is optimal). Our work certainly provides good insight into the multi-bidder problem, but there are still clear barriers. For example, in order to obtain revenue equal to the expected welfare, the auction must necessarily also maximize welfare. In our single-bidder model, this means that we can give away the item for free for $\Omega(T)$ rounds, but with multiple bidders, such careless behaviour would immediately make it impossible to achieve the optimal welfare. Regarding the mean-based revenue, while there is a natural generalization of our LP to multiple bidders, it's no longer clear how to achieve this revenue against conservative bidders, as all the relevant variables now implicitly depend on the actions of the other bidders. These are just examples of concrete barriers, and there are likely interesting conceptual barriers for this extension as well.

Another interesting direction is understanding the consequences of our work from the perspective of the buyer. Aside from certain corner configurations (e.g. the seller extracting the buyer's full welfare), it's not obvious how the buyer's utility changes. For instance, is it possible that the buyer's utility actually *increases* as the seller switches from the default strategy to the optimal mean-based revenue? Does the buyer ever benefit from using an "exploitable" learning strategy, so that the seller can exploit it and make them both happier?

Chapter 3

Multi-armed Bandit with Strategic Arms

The results of this chapter are based on joint work with Mark Braverman, Jon Schneider and Matt Weinberg [36].

3.1 Introduction

Classically, algorithms for problems in machine learning assume that their inputs are drawn either stochastically from some fixed distribution or chosen adversarially. In many contexts, these assumptions do a fine job of characterizing the possible behavior of problem inputs. Increasingly, however, these algorithms are being applied to contexts (ad auctions, search engine optimization, credit scoring, etc.) where the quantities being learned are controlled by rational agents with external incentives. To this end, it is important to understand how these algorithms behave in *strategic* settings.

The multi-armed bandit problem is a fundamental decision problem in machine learning that models the trade-off between exploration and exploitation, and is used extensively as a building block in other machine learning algorithms (e.g. reinforcement learning). A learner (who we refer to as the *principal*) is a sequential decision maker who at each time step t, must decide which of k arms to 'pull'. Pulling this arm bestows a reward (either adversarially or stochastically generated) to the principal, and the principal would like to maximize his overall reward. Known algorithms for this problem guarantee that the principal can do approximately as well as the best individual arm.

In this paper, we consider a strategic model for the multi-armed bandit problem where each arm is an individual strategic agent and each round one arm is pulled by an agent we refer to as the *principal*. Each round, the pulled arm receives a private reward $v \in [0, 1]$ and then decides what amount x of this reward gets passed on to the principal (upon which the principal receives utility x and the arm receives utility $v - x$). Each arm therefore has a natural tradeoff between keeping most of its reward for itself and passing on the reward so as to be chosen more frequently. Our goal is to design mechanisms for the principal which simultaneously learns which arms are valuable while also incentivizing these arms to pass on most of their rewards.

This model captures a variety of dynamic agency problems, where at each time step the principal must choose to employ one of K agents to perform actions on the principal's behalf, where the agent's cost of performing that action is unknown to the principal (for example, hiring one of K contractors to perform some work, or hiring one of K investors with external information to manage some money - the important feature being that the principal doesn't know exactly how much they will pay/receive/etc. until the job is done, and the agent has a lot of freedom to set this ex-post). In this sense, this model can be thought of as a multi-agent generalization of the principal-agent problem in contract theory when agents are allowed private savings (see Section 3.1.2 for references). The model also captures, for instance, the interaction between consumers (as the principal) and many sellers deciding how steep a discount to offer the consumers - higher prices now lead to immediate revenue, but offering better discounts than your competitors will lead to future sales. In all domains, our model aims to capture settings where the principal has little domain-specific or market-specific knowledge, and can really only process the reward they get for pulling an arm and not any external factors that contributed to that reward.

There are two "obvious" approaches to try and solve these problems: Option one is to treat it like a procurement auction and run a reverse second-price auction. This doesn't quite work, however, in the case where the agents don't initially know how much reward they'll generate, so some amount of learning needs to enter the picture for a solution to be viable. Using the contractor as a *toy* running example: the contractor will not initially know how much it costs her to work on your home, but after working on your home several times *they* will start to learn how much the next one will cost (you will only learn how much they charge you). In any case, one cannot simply treat it like an auctions problem and ignore learning completely.

The second "obvious" approach is just to treat it as a learning problem, and ignore incentives completely. In fact, one oft-cited motivation for considering adversarial rewards in bandit settings is that arms might be strategic. Indeed, this is because even if the arms' rewards are stochastic, the utility they strategically pass on to the principal is unlikely to follow any distribution. Algorithms like EXP3 which guarantee low-regret in adversarial settings then seem like the natural "pure learning" approach. Interestingly, our main "negative result" shows that *any* adversarial learning algorithm admits a really bad approximate Nash equilibrium (more details below).

So auctions alone cannot solve the problem, nor can learning alone. To complement our main negative result, we show that the right combination of auctions and learning yields a positive result: an algorithm such that all approximate Nash result in good utility for the principal. We now overview our results in more detail.

3.1.1 Our results

Low-regret algorithms are far from strategyproof

Many algorithms for the multi-armed bandit problem are designed to work in worst-case settings, where an adversary can adaptively decide the value of each arm pull. Here, algorithms such as EXP3 ([18]) guarantee that the principal receives almost as much as if he had only pulled the best arm. Formally, such algorithms guarantee that the principal experiences at most $O(\sqrt{T})$ regret over T rounds compared to any algorithm that only plays a single arm (when the adversary is oblivious).

Given these worst-case guarantees, one might naively expect low-regret algorithms such as EXP3 to also perform well in our strategic variant. It is important to note, however, that single arm strategies perform dismally in this strategic setting; if the principal only ever selects one arm, the arm has no incentive to pass along any surplus to the principal. In fact, we show that the objectives of minimizing adversarial regret and performing well in this strategic variant are fundamentally at odds.

Theorem 3.1.1. *Let M be a low-regret algorithm for the classic multi-armed bandit problem with adversarially chosen values. Then there exists an instance of the strategic multi-armed bandit problem and an $o(T)$-Nash equilibrium for the arms where a principal running M receives at most $o(T)$ revenue.*

While not immediately apparent from the statement of Theorem 3.1.1, these instances where low-regret algorithms fail are far from pathological; in particular, there is a problematic equilibrium for any instance where arm i receives a fixed reward v_i each round it is pulled, as long as the the gap between the largest and second-largest v_i is not too large (roughly 1/#arms).

Here we assume the game is played under a *tacit* observational model, meaning that arms can only observe which arms get pulled by the principal, but not how much value they give to the principal. In particular, this means that arms can achieve this equilibrium despite not communicating directly with each other and not observing the actions of the other arms. This rules out various sorts of "grim trigger" collusion strategies (similar to collusion that occurs in the setting of repeated auctions, see [178]), where arms agree on a protocol ahead of time and immediately defect as soon as one arm deviates from this protocol. (Indeed, in an *explicit* observational model, where arms can see both which arms get pulled and how much value they pass on, it is easy to show even stronger results via such strategies; see the full version of [36] for details).

Instead, the strategies in the equilibrium of Theorem 3.1.1 take the form of *market-sharing strategies*, where arms calibrate their actions so that they each get played some proportion (e.g. $1/K$) of the time while passing on little utility to the principal. For example, consider a simple instance of this problem with two strategic arms, where the principal is using the low-regret EXP3 algorithm, and where arm 1 always gets private reward 1 if pulled and arm 2 always gets private reward 0.8. By always reporting some value slightly larger than 0.8, arm 1 can incentivize the principal to almost always pull it in the long run. This gains arm 1 roughly 0.2 utility per round (and arm 2 nothing). On the other hand, if arm 1 and arm 2 never pass along any surplus to the principal, they will likely be played equally often, gaining arm 1 roughly 0.5 utility per round and arm 2 0.4 utility per round.

To show such a market-sharing strategy works for general low-regret algorithms, much more work needs to be done. The arms must be able to enforce an even split of the principal's pulls (as soon as the principal starts lopsidedly pulling one arm more often than the others, the remaining arms can defect and start reporting their full value whenever pulled). As long as the principal guarantees good performance in the non-strategic adversarial case (achieving $o(T)$ regret), we show that the arms can (at $o(T)$ cost to themselves, and without explicitly communicating) cooperate so that they are all played equally often.

Mechanisms for strategic arms with stochastic values

We next show that, in contrast to Theorem 3.1.1, it is in fact possible for the principal to extract positive values from the arms per round, if we do not restrict the principal to use an adversarial low-regret algorithm (and hence there is a price to being adversarial low-regret).

We consider a setting where each arm i's reward when pulled is drawn independently from some distribution D_i with mean μ_i (unknown to the principal). In this case the principal can extract the value of the second-best arm (which is the best possible, as we show in Lemma 3.4.2). In the below statement, we are using the term "truthful mechanism" quite loosely as shorthand for "strategy that induces a game among the arms where each arm has a dominant strategy."

Theorem 3.1.2 (restatement of Corollary 3.4.2)**.** *Let μ' be the second largest mean amongst the set of μ_is. Then there exists a truthful mechanism for the principal that guarantees revenue at least $\mu' T - o(T)$ when the arms are playing according to any $o(T)$-Nash equlibrium.*

The mechanism in Theorem 3.1.2 can be thought of as a combination of a second-price auction with the explore-then-exploit strategy from multi-armed bandits. The principal divides the time horizon into three "phases". In the first phase (of size $o(T)$), the principal begins by asking each arm i to simply report their value each round, thus allowing the principal to learn which arm is the most valuable. In the second phase (which comprises the vast majority of the rounds), the principal asks the most valuable arm (the arm with the highest mean in the first phase) to give him the second-largest mean worth of value per round. If this arm fails to comply in any round, the principal avoids picking this arm for the remainder of the rounds. Finally, in the third phase, the principal uses a proper scoring rule to recompensate all arms for reporting truthfully in the first phase. (A more detailed description of the mechanism can be seen in Mechanisms 2 and 3 in Section 3.4).

As an added bonus, we show that this mechanism has similar guarantees in the setting where some arms are strategic and some arms are non-strategic (and our mechanism does not know which arms are which).

Theorem 3.1.3 (restatement of Theorem 3.4.1)**.** *Let μ_s be the second largest mean amongst the means of the strategic arms, and let μ_n be the largest mean amongst the means of the non-strategic arms. Then there exists a truthful mechanism for the principal that guarantees (with probability $1-o(1/T)$) revenue at least $\max(\mu_s, \mu_n)T - o(T)$ when arms play according to any $o(T)$-Nash equilibrium.*

In particular, this implies that Mechanism 3 has low-regret in the classical *stochastic* multi-armed bandits setting, and so the adversarial aspect of the low-regret guarantees is actually essential for the proof of Theorems 3.1.1.

A fair critique of this mechanism is that most of the work of learning the distributions of the arms is offloaded to the beginning of the game. This is appealing because it makes it much feasible to "slide in" some auction design and scoring rules to handle incentives. It is an interesting problem whether learning can still be done adaptively over time in this model, as such a procedure would necessitate a much more sophisticated treatment of incentives; see Section 3.5 for further discussion.

3.1.2 Related work

The study of classical multi-armed bandit problems was initiated by [171], and has since grown into an active area of study. The most relevant results for our paper concern the existence of low-regret bandit algorithms in the adversarial setting, such as the EXP 3 algorithm ([18]), which achieves regret $\tilde{O}(\sqrt{KT})$. Other important results in the classical setting include the upper confidence bound (UCB) algorithm for stochastic bandits ([137]) and the work of [105] for Markovian bandits. For further details about multi-armed bandit problems, see the survey [45].

One question that arises in the strategic setting (and other adaptive settings for multi-armed bandits) is what the correct notion of regret is; standard notions of regret guarantee little, since the best overall arm may still have a small total reward. [13] considered the multi-armed bandit problem with an adaptive adversary and introduced the quantity of "policy regret", which takes the adversary's adaptiveness into account. They showed that any multi-armed bandit algorithm will get $\Omega(T)$ policy regret. This indicates that it is not enough to treat strategic behaviors as an instance of adaptively adversarial behavior; good mechanisms for the strategic multi-armed bandits problem must explicitly take advantage of the rational self-interest of the arms.

Our model bears some similarities to the principal-agent problem of contract theory, where a principal employs an more informed agent to make decisions on behalf of the principal, but where the agent may have incentives misaligned from the principal's interests when it gets private savings (for example [52]). For more details on principal-agent problem, see the book [136]. Our model can be thought of as a sort of multi-armed version of the principal-agent problem, where the principal has many agents to select from (the arms) and can try to use competition between the agents to align their interests with the principal.

Our negative results are closely related to results on collusions in repeated auctions. Existing theoretical work [147, 17, 126, 11, 12, 178] has shown that collusive schemes exist in repeated auctions in many different settings, e.g., with/without side payments, with/without communication, with finite/infinite typespace. In some settings, efficient collusion can be achieved, i.e., bidders can collude to allocate the good to the bidders who values it the most and leave 0 asymptotically to the seller. Even without side payments and communication, [178] showed that tacit collusion exists and can achieve asymptotic efficiency with a large cartel.

Our truthful mechanism uses a proper scoring rule [42, 148] implicitly. In general, scoring rules are used to assessing the accuracy of a probabilistic prediction. In our mechanisms, we use a logarithmic scoring rule to incentivize arms to truthfully report their average rewards.

Our setting is similar to settings considered in a variety of work on dynamic mechanism design, often inspired by online advertising. [26] considers the problem where a buyer wants to buy a stream of goods with an unknown value from two sellers, and examines Markov perfect equilibria in this model. [22, 72, 20] study truthful pay-per-click auctions where the auctioneer wishes to design a truthful mechanism that maximizes the social welfare. [133, 104] consider the scenario where the principal cannot directly choose which arm to pull, and instead must incentivize a stream of strategic players to prevent them from acting myopically. [9, 10] consider a setting where a seller repeatedly sells to a buyer with unknown value distribution, but the buyer is more heavily discounted than the seller. [127] develops a general method for finding optimal mechanisms in settings with dynamic private information. [158] develops an ex ante efficient mechanism for the Cost-Per-Action charging scheme in online advertising.

3.2 Our Model

3.2.1 Classic Multi-Armed Bandits

We begin by reviewing the definition of the classic multi-armed bandits problem and associated quantities.

In the classic multi-armed bandit problem a learner (the *principal*) chooses one of K choices (arms) per round, over T rounds. On round t, the principal receives some reward $v_{i,t} \in [0,1]$ for pulling arm i. The values $v_{i,t}$ are either drawn independently from some distribution corresponding to arm i (in the case of *stochastic bandits*) or adaptively chosen by an adversary (in the case of *adversarial bandits*). Unless otherwise specified, we will assume we are in the adversarial setting.

Let I_t denote the arm pulled by the principal at round t. The *revenue* of an algorithm M is the random variable

$$\mathsf{Rev}(M) = \sum_{t=1}^{T} v_{I_t,t}$$

and the the *regret* of M is the random variable

$$\mathsf{Reg}(M) = \max_i \sum_{t=1}^{T} v_{i,t} - \mathsf{Rev}(M)$$

Definition 3.2.1 (δ-Low Regret Algorithm). *Mechanism M is a δ-low regret algorithm for the multi-armed bandit problem if*

$$\mathrm{E}[\mathsf{Reg}(M)] \leq \delta.$$

Here the expectation is taken over the randomness of M and the adversary.

Definition 3.2.2 ((ρ, δ)-Low Regret Algorithm). *Mechanism M is a (ρ, δ)-low regret algorithm for the multi-armed bandit problem if with probability* $1 - \rho$*,*

$$\mathsf{Reg}(M) \leq \delta.$$

There exist $O(\sqrt{KT \log K})$-low regret algorithms and $(\rho, O(\sqrt{KT \log(K/\rho)}))$-low regret algorithms for the multi-armed bandit problem; see Section 3.2 of [45] for details.

3.2.2 Strategic Multi-Armed Bandits

The strategic multi-armed bandits problem builds upon the classic multi-armed bandits problem with the notable difference that now arms are strategic agents with the ability to withhold some payment from the principal. Instead of the principal directly receiving a reward $v_{i,t}$ when choosing arm i, now arm i receives this reward and passes along some amount $w_{i,t}$ to the principal, gaining the remainder $v_{i,t} - w_{i,t}$ as utility.

For simplicity, in the strategic setting, we will assume the rewards $v_{i,t}$ are generated stochastically; that is, each round, $v_{i,t}$ is drawn independently from a distribution D_i (where the distributions D_i are known to all arms but not to the principal). While it is possible to pose this problem in the adversarial setting (or other more general settings), this comes at the cost of there being no clear notion of strategic equilibrium for the arms.

This strategic variant comes with two additional modeling assumptions. The first is the informational model of this game; what information does an arm observe when some other arm is pulled. We define two possible observational models:

1. **Explicit:** After each round t, every arm sees the arm played I_t along with the quantity $w_{I_t,t}$ reported to the principal.

2. **Tacit:** After each round t, every arm only sees the arm played I_t.

In both cases, only arm i knows the size of the original reward $v_{i,t}$; in particular, the principal also only sees the value $w_{i,t}$ and learns nothing about the amount withheld by the arm. Collusion between arms is generally significantly easier in the explicit observational model than in the tacit observational model, and for this reason we will assume we are in the tacit observational model unless otherwise stated.

The second modeling assumption is whether to allow arms to go into debt while paying the principal. In the *restricted payment* model, we impose that $w_{i,t} \leq v_{i,t}$; an arm cannot pass along more than it receives in a given round. In the *unrestricted payment* model, we let $w_{i,t}$ be any value in $[0, 1]$. We prove our negative results in the restricted payment model and our positive results in the unrestricted payment model, but our proofs for our negative results work in both models (in particular, it is easier to collude and prove negative results in the unrestricted payment model) and Mechanism 3 can be adapted to work in the restricted payment model (see discussion in Section 3.4.2).

Finally, we proceed to define the set of strategic equilibria for the arms. We assume the mechanism M of the principal is fixed ahead of time and known to the K arms. If each arm i is using a (possibly adaptive) strategy S_i, then the expected utility of arm i is defined as

$$u_i(M, S_1, \dots, S_K) = \mathrm{E}\left[\sum_{t=1}^{T}(v_{i,t} - w_{i,t}) \cdot \mathbb{1}_{I_t=i}\right].$$

An ε-Nash equilibrium for the arms is then defined as follows.

Definition 3.2.3 (ε-Nash Equilibrium for the arms)**.** *Strategies $(S_1, ..., S_K)$ form an ε-Nash equilibrium for the strategic multi-armed bandit problem if for all $i \in [n]$ and any deviating strategy S_i',*

$$u_i(S_1, \dots, S_i, \dots, S_K) \geq u_i(S_1, \dots, S_i', \dots, S_K) - \varepsilon.$$

Similarly as before, the revenue of the principal in this case is the random variable

$$\mathsf{Rev}(M, S_1, \dots, S_K) = \sum_{t=1}^{T} w_{I_t,t}.$$

The goal of the principal is to choose a mechanism M which guarantees large revenue in any ε-Nash Equilibrium for the arms.

In Section 3.4, we will construct mechanisms for the strategic multi-armed bandit problem which are truthful for the arms. We define the related terminology below.

Definition 3.2.4 (Dominant Strategy)**.** *When the principal uses mechanism M, we say S_i is a dominant strategy for arm i if for any deviating strategy S_i' and any strategies for other arms $S_1, .., S_{i-1}, S_{i+1}, ..., S_K$,*

$$u_i(M, S_1, \dots, S_i, \dots, S_K) \geq u_i(M, S_1, \dots, S_i', \dots, S_K).$$

Definition 3.2.5 (Truthfulness)**.** *We say that a mechanism M for the principal is truthful, if all arms have some dominant strategies.*

3.3 Negative Results Overview

In this section we give a sketch of the proof of our main theorem, Theorem 3.3.1. The full list of our negative results and proofs can be found in the full version of [36].

Theorem 3.3.1. *Let mechanism M be a (ρ, δ)-low regret algorithm for the multi-armed bandit problem with K arms, where $K \leq T^{1/3}/\log(T)$, $\rho \leq T^{-2}$, and $\delta \geq \sqrt{T \log T}$. Then in the strategic multi-armed bandit problem under the tacit observational model, there exist distributions D_i and an $O(\sqrt{KT\delta})$-Nash Equilibrium for the arms where the principal gets at most $O(\sqrt{KT\delta})$ revenue.*

Proof Sketch. The underlying idea here is that the arms work to try to maintain an equal market share, where each of the K arms are each played approximately $1/K$ of the time. To ensure this happens, arms collude so that arms that aren't as likely to be pulled pass along more than arms that have been pulled a lot or are more likely to be pulled; this ends up forcing any low-regret algorithm for the principal to choose all the arms equally often. Interestingly, this collusion strategy is *mechanism dependent*, as arms need to estimate the probability they will be pulled in the next round.

More formally, let μ_i denote the mean of the ith arm's distribution D_i. Without loss of generality, further assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. We will show that as long as $\mu_1 - \mu_2 \leq \frac{\mu_1}{K}$, there exists some $O(\sqrt{KT\delta})$-Nash equilibrium for the arms where the principal gets at most $O(\sqrt{KT\delta})$ revenue.

We begin by describing the equilibrium strategy S^* for the arms. Let $c_{i,t}$ denote the number of times arm i has been pulled up to time t. Set $B = 7\sqrt{KT\delta}$ and set $\theta = \sqrt{\frac{K\delta}{T}}$. The equilibrium strategy for arm i at time t is as follows:

1. If at any time $s \leq t$ in the past, there exists an arm j with $c_{j,s} - c_{i,s} \geq B$, defect and offer your full value $w_{i,t} = \mu_i$.

2. Compute the probability $p_{i,t}$, the probability that the principal will pull arm i conditioned on the history so far.

3. Offer $w_{i,t} = \theta(1 - p_{i,t})$.

The main technical challenge in proving that this strategy is an equilibrium involves showing that, if all arms are following this strategy and the principal is using a low-regret mechanism, then with high probability the arms will not defect. Here the low-regret property of the mechanism M is essential (indeed, as our positive results imply, the theorem is not true without this assumption). In particular, by the construction of $w_{i,t}$ in terms of $p_{i,t}$, the principal's expected total regret (here defined to be the sum of the principal's regrets with respect to each arm) will increase each round by some amount proportional to the variance of the $p_{i,t}$. Intuitively, this implies that the values $p_{i,t}$ cannot be too far from uniform for too many rounds, and therefore that each arm should be picked approximately the same proportion of the time. This is formalized in the following lemma:

Lemma 3.3.1. *If all arms are using strategy S^*, then with probability $\left(1 - \frac{3}{T}\right)$, $|c_{i,t} - c_{j,t}| \leq B$ for all $t \in [T], i, j \in [K]$.*

Proof. As before, assume that all arms are playing the strategy S^* with the modification that they never defect. This does not change the probability that $|c_{i,t} - c_{j,t}| \leq B$ for all $t \in [T], i, j \in [K]$.

Define $R_{i,t} = \sum_{s=1}^{t} w_{i,s} - \sum_{s=1}^{t} w_{I_s,s}$ be the regret the principal experiences for not playing only arm i up until time t. We begin by showing that with probability at least $1 - \frac{2}{T}$, $R_{i,t}$ lies in $[-K\theta\sqrt{T \log T} - (K-1)\delta, \delta]$ for all $t \in [T]$ and $i \in [K]$.

To do this, first note that since the principal is using a (T^{-2}, δ)-low-regret algorithm, with probability at least $1 - T^{-2}$ the regrets $R_{i,t}$ are all upper bounded by δ at any fixed time t. Via the union bound, it follows that $R_{i,t} \leq \delta$ for all i and t with probability at least $1 - \frac{1}{T}$.

To lower bound $R_{i,t}$, we will first show that $\sum_{i=1}^{K} R_{i,t}$ is a submartingale in t. Note that, with probability $p_{j,t}$, $R_{i,t+1}$ will equal $R_{i,t} + \theta((1 - p_{j,t}) - (1 - p_{i,t}))$. We then have

$$
\begin{aligned}
\mathrm{E}\left[\sum_{i=1}^{K} R_{i,t+1} \middle| \sum_{i=1}^{K} R_{i,t}\right] &= \sum_{i=1}^{K} R_{i,t} + \sum_{i=1}^{K} p_{i,t} \sum_{j=1}^{K} \theta((1-p_{j,t}) - (1-p_{i,t})) \\
&= \sum_{i=1}^{K} R_{i,t} + \sum_{i=1}^{K} p_{i,t} \sum_{j=1}^{K} \theta(p_{i,t} - p_{j,t}) \\
&= \sum_{i=1}^{K} R_{i,t} + \theta \sum_{i=1}^{K} p_{i,t}(K p_{i,t} - 1) \\
&= \sum_{i=1}^{K} R_{i,t} + \theta \left(K \sum_{i=1}^{K} p_{i,t}^2 - \sum_{i=1}^{K} p_{i,t} \right) \\
&\geq \sum_{i=1}^{K} R_{i,t}
\end{aligned}
$$

where the last inequality follows by Cauchy-Schwartz. It follows that $\sum_{i=1}^{K} R_{i,t}$ forms a submartingale.

Moreover, note that (since $|p_i - p_j| \leq 1$) $|R_{i,t+1} - R_{i,t}| \leq \theta$. It follows that $\left|\sum_{i=1}^{K} R_{i,t+1} - \sum_{i=1}^{K} R_{i,t}\right| \leq K\theta$ and therefore by Azuma's inequality that, for any fixed $t \in [T]$,

$$
\Pr\left[\sum_{i=1}^{K} R_{i,t} \leq -2K\theta\sqrt{T \log T}\right] \leq \frac{1}{T^2}.
$$

With probability $1 - \frac{1}{T}$, this holds for all $t \in [T]$. Since (with probability $1 - \frac{1}{T}$) $R_{i,t} \leq \delta$, this implies that with probability $1 - \frac{2}{T}$, $R_{i,t} \in \left[-2K\theta\sqrt{T \log T} - (K-1)\delta, \delta\right]$.

We next proceed to bound the probability that $c_{i,t} - c_{j,t} > B$ for a i, j, and t. Define

$$
S_t^{(i,j)} = \left(c_{i,t} - c_{j,t} + \frac{1}{\theta}(R_{i,t} - R_{j,t}) \right).
$$

We claim that $S_t^{(i,j)}$ is a martingale. To see this, we first claim that $R_{i,t+1} - R_{j,t+1} = R_{i,t} - R_{j,t} - \theta(p_{i,t} - p_{j,t})$. Note that, if arm k is pulled, then $R_{i,t+1} = R_{i,t} + \theta((1 - p_{i,t}) -$

$(1-p_{k,t})) = R_{i,t} + \theta(p_{k,t} - p_{i,t})$ and similarly, $R_{j,t+1} = R_{j,t} + \theta(p_{k,t} - p_{j,t})$. It follows that $R_{i,t+1} - R_{j,t+1} = R_{i,t} - R_{j,t} - \theta(p_{i,t} - p_{j,t})$.

Secondly, note that (for any arm k) $\mathrm{E}[c_{k,t+1} - c_{k,t} | p_t] = p_{k,t}$, and thus $\mathrm{E}[c_{i,t+1} - c_{j,t+1} - (c_{i,t} - c_{j,t}) | p_t] = p_{i,t} - p_{j,t}$. It follows that

$$
\begin{aligned}
\mathrm{E}[S_{t+1}^{(i,j)} - S_t^{(i,j)} | p_t] &= \mathrm{E}[(c_{i,t+1} - c_{j,t+1}) - (c_{i,t} - c_{j,t}) | p_t] \\
&\quad + \frac{1}{\theta}\mathrm{E}[(R_{i,t+1} - R_{j,t+1}) - (R_{i,t} - R_{j,t}) | p_t] \\
&= (p_{i,t} - p_{j,t}) - (p_{i,t} - p_{j,t}) \\
&= 0
\end{aligned}
$$

and thus that $\mathrm{E}[S_{t+1}^{(i,j)} | S_t^{(i,j)}] = S_t^{(i,j)}$, and thus that $S_t^{(i,j)}$ is a martingale. Finally, note that $|S_{t+1}^{(i,j)} - S_t^{(i,j)}| \le 2$, so by Azuma's inequality

$$
\Pr\left[S_t^{(i,j)} \ge 4\sqrt{T\log(TK)}\right] \le (TK)^{-2}
$$

Taking the union bound, we find that with probability at least $1 - \frac{1}{T}$, $S^{(i,j)} \le 4\sqrt{T\log(TK)}$ for all i, j, and t. Finally, since with probability at least $1 - \frac{2}{T}$ each $R_{i,t}$ lies in $\left[-2K\theta\sqrt{T\log T} - (K-1)\delta, \delta\right]$, with probability at least $1 - \frac{3}{T}$ we have that (for all i, j, and t)

$$
\begin{aligned}
c_{i,t} - c_{j,t} &= S_t^{(i,j)} - \frac{1}{\theta}(R_{i,t} - R_{j,t}) \\
&\leq 4\sqrt{T\log(TK)} + \frac{1}{\theta}\left|R_{i,t} - R_{j,t}\right| \\
&\leq 4\sqrt{T\log(TK)} + 2K\sqrt{T\log T} + \frac{K\delta}{\theta} \\
&\leq \frac{7K\delta}{\theta} \\
&= 7K\sqrt{T\delta} \\
&= B
\end{aligned}
$$

□

Lemma 3.3.1 implies that if each arm plays strategy S^*, then each arm i will receive on average μ_i/K per round. To finish the proof, it suffices to note that by deviating and playing a different strategy S from S^*, one of two things can occur. If playing this different strategy S does not trigger the defect condition in (1), then still each arm will be played roughly $1/K$ of the time (and your total utility is unchanged up to $o(T)$ additive factors). On the other hand, once the defect condition is triggered, you can receive at most $\mu_1 - \mu_2$ utility per round (and only if you are arm 1). This implies that as long as $\mu_1/K > \mu_1 - \mu_2$, there is no incentive to deviate.

□

While the theorem above merely claims that a bad set of distributions for the arms exists, the proof shows it is possible to collude in a wide range of instances - in particular, any collection of distributions which satisfies $\mu_1 - \mu_2 \leq \mu_1/K$. A natural question is whether we can extend the above results to show that it is possible to collude in any set of distributions.

One issue with the collusion strategy in the above proof is that if $\mu_1 - \mu_2 > \mu_1/K$, then arm 1 will have an incentive to defect in any collusive strategy that plays all the arms evenly (arm 1 can report a bit over μ_2 per round, and make $\mu_1 - \mu_2$ every round instead of μ_1 every

K rounds). One solution to this is to design a collusive strategy that plays some arms more than others in equilibrium (for example, playing arm 1 90% of the time). We show how to modify our result for two arms to achieve an arbitrary market partition and thus work over a broad set of distributions.

Theorem 3.3.2. *Let mechanism M be a (ρ, δ)-low regret algorithm for the multi-armed bandit problem with two arms, where $\rho \leq T^{-2}$ and $\delta \geq \sqrt{T \log T}$. Then, in the strategic multi-armed bandit problem under the tacit observational model, for* any *distributions D_1, D_2 of values for the arms (supported on $[\sqrt{\delta/T}, 1]$), there exists an $O(\sqrt{T\delta})$-Nash Equilibrium for the arms where a principal using mechanism M gets at most $O(\sqrt{T\delta})$ revenue.*

Unfortunately, it as not as easy to modify the proof of Theorem 3.3.1 to prove the same result for K arms. It is an interesting open question whether there exist collusive strategies for K arms that can achieve an arbitrary partition of the market.

3.4 Positive Results

In this section we will show that, in contrast to the previous results on collusion, there exists a mechanism for the principal that can obtain $\Theta(T)$ revenue from the arms when they play according to an $o(T)$-Nash equilibrium.

We begin by demonstrating a simpler version of our mechanism (Mechanism 2) that guarantees the principal $\Theta(T)$ revenue whenever the arms play according to their dominant strategies. In Section 3.4.2, we then show how to make this mechanism more robust (Mechanism 3) so that the principal is guaranteed $\Theta(T)$ revenue whenever the arms play according to any $o(T)$-approximate Nash equilibrium (thus showing a separation between the power of adversarial low-regret algorithms and general learning algorithms in this model). As an added bonus, we show that this mechanism also works for a combination of strategic and non-strategic arms (and therefore achieves low regret in the classical stochastic multi-armed bandits setting).

Throughout this section we will assume we are working in the tacit observational model and the unrestricted payment model (unless otherwise specified). All the proofs can be found in the full version of [36].

3.4.1 Good dominant strategy equilibria

This mechanism essentially incentivizes each arm to report the mean of its distribution and then runs a second-price auction, asking the arm with the highest mean for the second-highest mean each round.

Define μ_i as the mean of distribution D_i for $i = 1, \ldots, K$, let $\mu_{min} = \min_{i:\mu_i \neq 0}(\mu_i)$, and $u = -\log \mu_{min} + 1$. We assume throughout that $u = o(T/K)$.

2 Truthful mechanism for strategic arms with known stochastic values in the tacit model

Play each arm once (i.e. play arm 1 in the first round, arm 2 in the second round, etc.). Let w_i be the value arm i reports in round i.
Let $i^* = \arg\max w_i$ (breaking ties lexicographically), and let $w' = \max_{i \neq i^*} w_i$.
Tell arm i^* the value of w'. Play arm i^* for $R = T - (u+2)K - 1$ rounds. If arm i^* ever reports a value different from w', stop playing it immediately. If arm i^* always gives w', play it for one bonus round (ignoring the value it reports).
For each arm i such that $i \neq i^*$, play it for one round.
For each arm i satisfying $u + \log(w_i) \geq 0$, play it $\lfloor u + \log(w_i) \rfloor$ times. Then, with probability $u + \log(w_i) - \lfloor u + \log(w_i) \rfloor$, play arm i for one more round.

We will first show that the dominant strategy of each arm in this mechanism includes truthfully reporting their mean at the beginning, and then then compute the principal's revenue under this dominant strategy.

Lemma 3.4.1. *The following strategy is the dominant strategy for arm i in Mechanism 2:*

1. *(line 1 of Mechanism 2) Report the mean value μ_i of D_i the first time when arm i is played.*

2. *(lines 3,4 of Mechanism 2) If $i = i^*$, for the R rounds that the principal expects to see reported value w', report the value w'. For the bonus round, report 0. If $i \neq i^*$, report 0.*

3. (line 5 of Mechanism 2) For all other rounds, report 0*.*

Corollary 3.4.1. *Under Mechanism 2, the principal will receive revenue at least* $\mu' T - o(T)$ *when arms use their dominant strategies, where* μ' *is the second largest mean in the set of means* μ_i*.*

We additionally show that the performance of Mechanism 2 is as good as possible; no mechanism can do better than the second-best arm in the worst case.

Lemma 3.4.2. *Let* μ *and* μ' *be the largest and second largest values respectively among the* μ_i*. Then for any constant* $\alpha > 0$*, no truthful mechanism can guarantee* $(\alpha\mu + (1-\alpha)\mu')T$ *revenue in the worst case.*

3.4.2 Good approximate Nash equilbria

One issue with Mechanism 2 is that, while the principal achieves $\Theta(T)$ revenue when the arms play according to their dominant strategies, there can exist ϵ-Nash equilibria for the arms which still leave the principal with negligible revenue. For instance, if there are two arms with equal means $\mu_1 = \mu_2 = \mu$, one possible ϵ-Nash equilibrium is for them both to bid μ, and then for arm i^* to immediately defect after it is chosen. This is not a dominant strategy, since arm i^* surrenders its bonus for not defecting, but since this bonus is at most 1, this is still an ϵ-Nash equilibrium for any $\epsilon = o(T)$ which is larger than 1.

We can make Mechanism 3 more robust to strategies like this by increasing the size of the bonus with ϵ. If we additionally allow a tiny buffer between the current reported average and w', this mechanism has the added property that it works even when there are a mixture of strategic and non-strategic arms (and the principal does not know which are which). In particular, this Mechanism 3 obtains low-regret in the classical stochastic multi-armed bandits setting, which implies that our negative results in Section 3.3 are really due to the adversarial nature of the low-regret guarantees.

As before, define μ_i as the mean of distribution D_i for $i = 1, ..., K$. Our mechanism takes in two parameters, B (representing the size of the bonus) and M (representing the size of the buffer). We will set $B = 2\epsilon^{1/4}T^{3/4}/\mu_{min}$ and $M = 8B^{-1/2}\ln(KT)$. In addition, we will define $u = -\log(\min_{i:\mu_i \neq 0} \mu_i) + 2 + M$. We assume $u = o(\frac{T}{BK})$.

3 Truthful mechanism for strategic/non-strategic arms in the tacit model

Play each arm B times (i.e. play arm 1 in the first B rounds, arm 2 in the next B rounds, etc.). Let $\bar{w}_i$ be the average value arm i reported in its B rounds.
Let $i^* = \arg\max \bar{w}_i$ (breaking ties lexicographically), and let $w' = \max_{i \neq i^*} \bar{w}_i$.
Tell arm i^* the value of w'. Play arm i^* for $R = T - (u+3)BK$ rounds. If arm i^* ever reports values with average less than $w' - M$ in any round after B rounds in this step, stop playing it immediately. If arm i^* gives average no less than $w' - M$, play it for B bonus rounds (ignoring the value it reports).
For each arm i such that $i \neq i^*$, play it for B rounds.
For each arm i satisfying $u + \log(\bar{w}_i - M) \geq 0$, play it $B\lfloor(u + \log(\bar{w}_i - M))\rfloor$ times. Then, with probability $u + \log(\bar{w}_i - M) - \lfloor u + \log(\bar{w}_i - M)\rfloor$, play arm i for B more rounds.

We begin by characterizing the dominant strategy for Mechanism 3. Similarly as in Lemma 3.4.1, we show that this dominant strategy involves each arm reporting their true mean in the beginning rounds.

Lemma 3.4.3. *The following strategy is the dominant strategy for arm i in Mechanism 3:*

1. *(line 1 of Mechanism 3) For the first B rounds, report a total sum of* $(\mu_i + M)B$.

2. *(lines 3,4 of Mechanism 3) If* $i = i^*$*, for the R rounds that the principal expects to see reported value* w'*, report the value* $w' - M$*. For the B bonus rounds, report 0. If* $i \neq i^*$*, report 0.*

3. *(line 5 of Mechanism 3) For all other rounds, report* 0.

We use this to show that under any $o(T)$-Nash equilibrium, the principal receives $\mu' T - o(T)$ revenue under Mechanism 3.

Corollary 3.4.2. *Under Mechanism 3, the principal will receive revenue at least* $\mu' T - o(T)$ *whenever arms play according to an* ϵ*-Nash equilibrium, where* μ' *is the second largest mean in the set of means* μ_i *and* $\epsilon = o(T)$.

The dominant strategy in Lemma 3.4.3, as written, requires the arms to know their own means μ_i (in particular for step 1). However, if the arms don't initially know their means, they can instead simply report their value (plus M) each round, and still report a total sum of $(\mu_i + M)B$ in expectation. This no longer results in a strictly dominant strategy, but instead an $o(T)$-dominant strategy.

Lemma 3.4.4. *The following strategy is a prior-independent $o(T)$-dominant strategy for arm i in Mechanism 3:*

1. *(line 1 of Mechanism 3) For each round t in the first B rounds, report $v_{i,t} + M$.*

2. *(lines 3,4 of Mechanism 3) If $i = i^*$, for the R rounds that the principal expects to see reported value w', report the value $w' - M$. For the B bonus rounds, report 0. If $i \neq i^*$, report 0.*

3. *(line 5 of Mechanism 3) For all other rounds, report* 0.

It is an interesting question whether a more clever stochastic bandit algorithm can be embedded without destroying dominant strategies, and also whether a solution exists in exact dominant strategies for this model.

Similarly, the dominant strategy in Lemma 3.4.3 assumes we are in the unrestricted payment regime, because sometimes the value you must report (whether it is $\mu_i + M$ or $w' - M$) might be larger than the value received in that round. However, again it is possible to adapt the mechanism (by setting $M = 0$) and dominant strategy in Lemma 3.4.3 to work for arms in the restricted payment regime at the cost of transforming it into a $o(T)$-dominant strategy. To do this, arms (as in the previous paragraph) simply report their value each round in the first phase of the mechanism. In the second phase of the mechanism, instead of reporting w' each round, they again report their full value, until they have reported a total of Rw' (at which point they start reporting 0 for the rest of the game).

Finally, we consider the case when some arms are strategic and other arms are non-strategic. Importantly, the principal does not know which arms are strategic and which are non-strategic. We show in this case that the principal can get (per round) the larger of the largest mean of the non-strategic arms and the second largest mean of the strategic arms.

Theorem 3.4.1. *If the strategic arms all play according to in Lemma 3.4.3, then the principal will get at least* $\max(\mu_s, \mu_n)T - o(T)$ *with probability* $1 - o(1/T)$*. Here* μ_s *is the second largest mean of the strategic arms and* μ_n *is the largest mean of the non-strategic arms.*

3.5 Conclusions and Future Directions

We consider the multi-armed bandit problem with strategic arms: arms obtain a reward when pulled and may pass any of it onto the principal. Our first main result shows that treating this purely as a learning problem results in undesirable approximate Nash equilibria for the principle (guaranteeing only $o(T)$ reward over T rounds). Our second main result shows that a careful combination of auctions, learning, and scoring rules provides a learning algorithm such that every approximate Nash equilibrium guarantees the principal $\Omega(T)$ reward (and even better - the arms have a dominant strategy). Still, we are far from understanding the complete picture of multi-armed bandit problems in strategic settings. Many questions remain, both in our model and related models.

One limitation of our negative results is that they only show there exists some 'bad' approximate Nash equilibrium for the arms, i.e., one where any low-regret principal receives little revenue. This, however, says nothing about the space of all approximate Nash equilibria. Does there exist a low-regret mechanism for the principal along with an approximate Nash equilibria for the arms where the principal extracts significant utility? An affirmative answer to this question would raise hope for the possibility of a mechanism that can perform well in both the adversarial and strategic setting, whereas a negative answer would strengthen our claim that these two settings are fundamentally at odds.

One limitation of our positive results is that all of the learning takes place at the beginning of the protocol. As a result, our mechanism fails in cases where the arms' distributions can change over time. Is it possible to design good mechanisms for such settings? Ideally, any good mechanism should learn the arms' values continually throughout the T rounds, but accommodating this would require novel tools to handle incentives.

Throughout this paper, whenever we consider strategic bandits we assume their rewards are stochastically generated. Can we say anything about strategic bandits with adversarially generated rewards? The key barrier here seems to be defining what a strategic equilibrium is in this case - arms need some underlying priors to reason about their future expected utility.

Finally, there are other quantities one may wish to optimize instead of the utility of the principal. For example, is it possible to design an efficient principal, who almost always picks the best arm (even if the arm passes along little to the principal)? Theorem 3.3.1 implies the answer is no if the principal also has to be efficient in the adversarial case, but are there other models where we can answer this question affirmatively?

Part II

Truthful Mechanisms in Combinatorial Auctions

Chapter 4

On Simultaneous Two-player Combinatorial Auctions

The results of this chapter are based on joint work with Mark Braverman and Matt Weinberg [39].

4.1 Introduction

We consider the following communication problem: Alice and Bob each have some valuation functions $v_1(\cdot)$ and $v_2(\cdot)$ over subsets of m items, and their goal is to partition the items into $S, \bar{S}$ in a way that maximizes the *welfare*, $v_1(S)+v_2(\bar{S})$. We study both the *allocation problem*, which asks for a welfare-maximizing partition and the *decision problem*, which asks whether or not there exists a partition guaranteeing certain welfare, for binary XOS valuations. For interactive protocols with poly(m) communication, a tight 3/4-approximation is known for both [97, 82].

For interactive protocols, the allocation problem is provably *harder* than the decision problem: any solution to the allocation problem implies a solution to the decision problem with one additional round and $\log m$ additional bits of communication via a trivial reduction. Surprisingly, the allocation problem is provably *easier* for simultaneous protocols. Specifically, we show:

- There exists a simultaneous, randomized protocol with polynomial communication that selects a partition whose expected welfare is at least 3/4 of the optimum. This matches the guarantee of the best interactive, randomized protocol with polynomial communication.

- For all $\varepsilon > 0$, any simultaneous, randomized protocol that decides whether the welfare of the optimal partition is ≥ 1 or $\leq 3/4 - 1/108 + \varepsilon$ correctly with probability $> 1/2 + 1/\text{poly}(m)$ requires exponential communication. This provides a separation between the attainable approximation guarantees via interactive (3/4) versus simultaneous ($\leq 3/4 - 1/108$) protocols with polynomial communication.

In other words, this trivial reduction from decision to allocation problems provably requires the extra round of communication. We further discuss the implications of our results for the design of truthful combinatorial auctions in general, and extensions to general XOS valuations. In particular, our protocol for the allocation problem implies a new style of truthful mechanisms.

Intuitively, search problems (find the optimal solution) are considered "strictly harder" than decision problems (does a solution with quality $\geq Q$ exist?) for the following (formal) reason: once you find the optimal solution, you can simply evaluate it and check whether its quality is $\geq Q$ or not. The same intuition carries over to approximation as well: once you find a solution whose quality is within a factor α of optimal, you can distinguish between cases where solutions with quality $\geq Q$ exist and those where all solutions have quality $\leq \alpha Q$. The easy conclusion one then draws is that the communication (resp. runtime) required for an α-approximation to any decision problem is upper bounded by the communication (resp. runtime) required for an α-approximation to the corresponding search problem plus the communication (resp. runtime) required to evaluate the quality of a proposed solution.

Note though that for communication problems, in addition to the negligible increase in communication (due to evaluating the quality of the proposed solution), this simple reduction might also require (at least) an extra round of communication (because the parties can evaluate a solution's quality only after it is found). Still, it seems hard to imagine that this extra round is really necessary, and that somehow protocols exist that guarantee an (approximately) optimal solution without (approximately) learning their quality. The surprising high-level takeaway from our main results is that *this extra round of communication is provably necessary*: Theorems 4.1.1 and 4.1.2 provide a natural communication problem (combinatorial auctions) such that a 3/4-approximation for the search problem can be found by a simultaneous protocol[1] with polynomial communication, but every simultaneous prootcol guaranteeing a $(3/4-1/108+\varepsilon)$-approximation for the decision problem requires $\exp(m)$ communication.

At this point, we believe our results to have standalone interest, regardless of how we wound up at this specific communication problem. But there is a rich history related to the design of truthful combinatorial auctions motivating our specific question, which we overview below.

4.1.1 Combinatorial Auctions - how did we get here?

In a combinatorial auction, a designer with m items wishes to allocate them to n bidders so as to maximize the *social welfare*. That is, if bidder i has a monotone valuation function $v_i : 2^{[m]} \to \mathbb{R}_+$,[2] the designer wishes to find disjoint sets $S_1, \ldots, S_n$ maximizing $\sum_i v_i(S_i)$. The history of combinatorial auctions is rich, and the problem has been considered with and without incentives, with and without Bayesian priors, and in various models of computation (see Section 4.1.5 for brief overview). The overarching theme in all of these works is to

[1] A simultaneous protocol has one round of communication: Alice and Bob each simultaneously send a message and then no further communication takes place.

[2] By monotone, we mean that $v_i(S) \geq v_i(T)$ for all $T \subseteq S$.

try and answer the following core question: *Are truthful mechanisms as powerful as (not necessarily truthful) algorithms?*[3]

For many instantiations of the above question, the answer is surprisingly yes. For example, without concern for computational/communication complexity, the celebrated Vickrey-Clarke-Groves auction is a truthful mechanism that always selects the welfare-maximizing allocation (and therefore achieves welfare equal to that of the best algorithm) [187, 63, 110]. Of course, the welfare maximization problem is NP-hard and also requires exponential communication between the bidders, even to guarantee a $1/\sqrt{m}$-approximation. A poly-time algorithm (with polynomial communication) is known to match this guarantee [168, 131, 43], and interestingly, a poly-time truthful mechanism (with polynomial communication) was later discovered as well [139].

The state of affairs gets even more interesting if we restrict to proper subclasses of monotone valuations such as *submodular* valuations.[4] Here, a very simple greedy algorithm is known to find a 1/2-approximation in both poly(n, m) black-box value queries to each $v_i(\cdot)$, and polynomial runtime (in n, m, and the description complexity of each $v_i(\cdot)$) [140], and a series of improvements provides now a (1-1/e)-approximation, which is tight [188, 153, 83]. Yet, another series of works also proves that any truthful mechanism that runs in polynomial time (in n, m, and the description complexity of each $v_i(\cdot)$), or makes only poly(n, m) black-box value queries to each $v_i(\cdot)$ achieves at best an $1/m^{\Omega(1)}$-approximation [67, 75, 87, 84]. So while poly-time algorithms, or algorithms making poly(n, m) black-box value queries can achieve constant-factor approximations, poly-time truthful mechanisms and truthful mechanisms making poly(n, m) black-box value queries can only guarantee an $1/m^{\Omega(1)}$-approximation, and there is a separation.

But this is far from the whole story: already ten years ago, quite natural truthful mechanisms were developed that achieved an $1/O(\log^2 m)$-approximation [81], which were subse-

[3]Note that combinatorial auctions is not the only literature to study this question, see Section 6.1.1 for very brief discussion of other examples such as combinatorial public projects [165] and job scheduling [161]. We just note here that combinatorial auctions remain the core testbed for this line of work.

[4]A function is submodular if $v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$.

quently improved to $1/O(\sqrt{\log m})$ [74, 134, 76], and even hold for the much broader class of XOS valuations.[5] As these approximation guarantees are better than the lower bounds referenced in the previous paragraph, it seems that perhaps there should be some kind of contradiction: any reasonable definition of "natural" should imply "poly-time," right? The catch is that each of these mechanisms are essentially posted-price mechanisms: they (essentially) offer each bidder a price p_j for item j, and let the buyer choose any subset of items they want to purchase. These prices can be computed in poly-time, but the barrier is that deciding which subset of items the bidder wishes to purchase, called a *demand query*, is in general NP-hard (assuming a succinct representation of the valuation function is given), or requires exponentially many black-box value queries. So the only reason these mechanisms don't fall victim to the strong lower bounds of the previous paragraph is because they get to ask each bidder to compute a single demand query, and this query is used to select exactly the set of items that bidder receives.

The point is that while these existing separations are major results, and rule out certain classes of natural truthful mechanisms from achieving desirable approximation ratios, they are perhaps not addressing "the right" model if posted-price mechanisms with poly-time computable prices provide approximation guarantees that significantly outperform known lower bounds. Therefore, it seems that communication is really the right complexity measure to consider, if one wants the resulting lower bounds to hold against all "natural" mechanisms. Unfortunately, the state-of-affairs for communication complexity of combinatorial auctions lags pretty far behind the aforementioned complexity measures. For instance, existing literature doesn't provide a single lower bound against truthful mechanisms that doesn't also hold against algorithms. That is, wherever it's known that no truthful mechanism with communication at most C obtains an approximation ratio better than α when buyers have valuations in class V, it's because it's also known that no algorithm/protocol with communication at

[5] A valuation is XOS if there exists a matrix of item valuations v_{ij} and $v_i(S) = \max_j\{\sum_{i \in S} v_{ij}\}$. XOS valuations are also called fractionally subadditive, and are a proper subclass of subadditive valuations (where $v(S \cup T) \leq v(S) + v(T)$).

most C obtains an approximation ratio better than α when buyers have valuations in class V. On the other hand, the best known truthful mechanisms with polynomial communication for (say) XOS bidders achieve an $1/O(\sqrt{\log m})$ approximation [76], while the best known algorithms with polynomial communication obtain a $(1-1/e)$-approximation [82, 97, 98]. Even for the case of just two bidders, the best known truthful mechanisms with polynomial communication achieve a 1/2-approximation (which is trivial - just give the grand bundle of all items to whoever values it most), while the best known algorithms with polynomial communication achieve a 3/4-approximation (which is tight). It's fair to say that determining whether or not there's a separation in what approximation guarantees are possible for algorithms with polynomial communication and truthful mechanisms with polynomial communication for any class of valuations between submodular and subadditive is one of the core, concrete open problems in Algorithmic Mechanism Design.

Progress on this front had largely been stalled until very recent work of Dobzinski provided a clear path to possibly proving a separation (and it seems to be an accepted conjecture that indeed a separation exists) [77]. Without getting into details of the complete result, one implication is the following: if there exists a truthful mechanism with polynomial communication for 2-player combinatorial auctions with XOS (/submodular/subadditive) valuations that guarantees an approximation ratio of α, then there exists a *simultaneous* protocol with polynomial communication for 2-player combinatorial auctions with XOS (/submodular/subadditive) valuations that guarantees an approximation ratio of α as well. Let us emphasize this point again: in general, interactive protocols with polynomial communication *do not* imply simultaneous protocols with polynomial communication, and numerous well-known problems have polynomial interactive protocols, but require exponential simultaneous communication [166, 88, 162, 79, 8, 16]. But, Dobzinski's result asserts that because of the extra conditions on *truthful* (interactive) mechanisms, their existence indeed implies a simultaneous (not necessarily truthful) protocol of comparable communication complexity. So "all" one has to do to prove lower bounds against truthful mechanisms for 2-player combinatorial auctions is prove lower bounds against simultaneous protocols, motivating the study of simultaneous 2-player combinatorial auctions.

At first glance, it perhaps seems obvious that achieving strictly better than a 1/2-approximation via a simultaneous protocol should be impossible, and it's just a matter of finding the right tools to prove it.[6] This is because quite strong lower bounds are known for "sketching" valuation functions, that is, finding a succinct representation of a function that allows for *approximate* evaluation of value queries. For example, it's known that any sketching scheme for XOS valuations that allows for evaluation of value queries to be accurate within a $o(m)$-factor requires superpoly(m) size [23]. So if somehow a $1/(2-\varepsilon)$-approximation could be guaranteed with a poly(m)-communication simultaneous protocol, it is *not* because enough information is transmitted to evaluate value queries within any non-trivial error. At first glance, it perhaps seems unlikely that such a protocol can possibly exist. Surprisingly, our work shows not only that a $1/(2-\varepsilon)$-approximation is achievable with poly(m) simultaneous communication, but (depending on exactly the question asked) poly(m) simultaneous communication suffices to achieve the same approximation guarantees as the best possible interactive protocol with poly(m) communication.

4.1.2 Simultaneous Protocols for Welfare Maximization

In this work, we specifically study the welfare maximization problem for two bidders with *binary XOS valuations*.[7] Binary XOS valuations are a natural starting point since welfare maximization is especially natural when phrased as a communication problem. Depending on whether one wants to decide the quality of the welfare-optimal allocation, or actually find an allocation inducing the optimal welfare, welfare maximization for binary XOS bidders is equivalent to one of the following:[8]

[6]Indeed, that is what the authors conjectured at the onset of this work.

[7]A function is binary XOS if all v_{ij} in the matrix representation are 0 or 1.

[8]Equivalent definitions are given in Section 4.1.4 which are stated more in the language of welfare maximization. We pose these statements here since these formulations make for an especially natural communication problem.

Definition 4.1.1 (BXOS Decision Problem)**.** *Alice is given as input a subsets of* $[m]$*,* $A_1, \dots, A_a$*. Bob is given as input b subsets of* $[m]$*,* $B_1, \dots, B_b$*, and both see input* X*. Determine whether or not there exists an* i, j *such that* $|A_i \cup B_j| \geq X$*. A protocol is an* α*-approximation if whenever there exists an* i, j *such that* $|A_i \cup B_j| \geq X$*, it answers yes, and whenever* $\max_{i,j}\{|A_i \cup B_j|\} < X/\alpha$ *it answers no, but may have arbitrary behavior in between.*

Definition 4.1.2 (BXOS Allocation Problem)**.** *Alice is given as input a subsets of* $[m]$*,* $A_1, \dots, A_a$*. Bob is given as input b subsets of* $[m]$*,* $B_1, \dots, B_b$*. Output a partition of items* $S, \bar{S}$ *maximizing* $\max_{i,j}\{|A_i \cap S| + |B_j \cap \bar{S}|\}$ *(over all partitions).*[9]

Recall that typically we think of decision problems as being "easier" than allocation/search problems: certainly if you can find a welfare maximizing allocation, you can also determine its welfare (and this claim is formal for interactive protocols with $\text{poly}(m)$ communication). Our main result asserts that this intuition breaks down for simultaneous protocols: the decision problem is strictly harder than the allocation/search problem. To the best of our knowledge, this is the first instance of such a separation.

Theorem 4.1.1. *There exists a randomized, simultaneous protocol with* $\text{poly}(m)$ *communication that obtains a* $3/4$*-approximation for the BXOS allocation problem. This is the best possible, as even randomized, interactive protocols require* $2^{\Omega(m)}$ *communication to do better.*

Theorem 4.1.2. *For all* $\varepsilon > 0$*, any randomized, simultaneous protocol that obtains a* $(3/4 - 1/108 + \varepsilon)$*-approximation for the BXOS decision problem with probability larger than* $1/2 + 1/\text{poly}(m)$ *requires* $2^{\Omega(m)}$ *communication.*

Future sections contain more precise versions (that reference the protocols achieving them) of Theorems 4.1.1 (Theorem 4.4.1) and 4.1.2 (Theorem 4.5.1).

[9] A protocol is an α-approximation if it outputs a partition $S, \bar{S}$ guaranteeing $\alpha \cdot \max_{i,j}\{|A_i \cap S| + |B_j \cap \bar{S}|\} \geq \max_{i,j,T}\{|A_i \cap T| + |B_j \cap T|\}$.

4.1.3 Extensions and Implications for Truthful Combinatorial Auctions

Part of the analysis of our protocols actually makes use of the binary assumption (as opposed to holding for general XOS). Part of the analysis, however, does not. In particular, our same protocols when applied to general XOS functions yield a deterministic, simultaneous $(3/4 - 1/32 - \varepsilon)$-approximation for both problems, and a deterministic 2-round $(3/4 - \varepsilon)$-approximation for both problems for general XOS functions.

We are also able to show that a modification of our protocol yields a 1/2-approximation for any number of binary XOS bidders, and that this protocol implies a *strictly* truthful mechanism.[10] The mechanism is quite different from existing approaches, and could inspire better truthful mechanisms in domains where previous molds provably fail. Essentially, the designer offers a menu of lotteries to each bidder and the cost of each lottery depends on how "flexible" the option is. So for instance, taking item one deterministically will be more expensive than taking a single item uniformly at random. The pricing scheme is designed exactly so that each bidder is strictly incentivized to follow our simultaneous protocol.

Finally, while our results have standalone merit outside the scope of truthful combinatorial auctions, it is important to properly quantify their impact in this direction. Dobzinski's recent reduction shows that truthful combinatorial auctions with polynomial communication imply simultaneous algorithms for the *allocation problem*. So Theorem 4.1.2 *does not* rule out the possibility of a truthful mechanism for two XOS bidders that requires polynomial communication and guarantees a 3/4-approximation (more on this in Section 4.6).

4.1.4 Brief Preliminaries and Roadmap

Below we give some brief preliminaries. Section 4.2 provides a toy setting to help develop intuition for where the gap between allocation and decision problem comes from. Section 4.3

[10] By this we mean it is a *strongly* dominant strategy for bidders to follow the protocol, and not just that they are indifferent between following and not following.

provides a warmup for our protocols via a 2/3-approximation for the allocation problem and a 3/5-approximation for the decision problem. Sections 4.4 and 4.4.1 contain our positive results, and Section 4.5 contains details on our lower bound.

In a combinatorial auction, there are n players and m items. In 2-party case, we call the first player Alice and the second player Bob. Each player i has a valuation function $v_i : 2^{[m]} \to \mathbb{R}^+$. (We require $v_i(\emptyset) = 0$.) The goal for the auctioneer is to find an allocation $S_1, ..., S_n$ ($S_1 \cap \cdots \cap S_n = \emptyset$) to maximize the social welfare $\sum_{i=1}^{n} v_i(S_i)$.

1. When we use "**protocol**", it means that players honestly follow the protocol and the challenge is to make the protocol have good approximation ratio, polynomial communication cost and possibly small number of rounds. In this paper, we use the standard communication complexity model and we allow public randomness and private randomness. For details, we refer the reader to [135]. We want to emphasize two relevant properties of the communication protocols here:

    (a) We care about the number of rounds of a protocol. In each round, all the messages need to be sent simultaneously. We use "simultaneous protocols" to denote protocols with only one round of communication.

    (b) All the protocols discussed in this paper are in the "blackboard model." In the blackboard model, each message is broadcasted. Or in other words, each message is written on a blackboard for all players and the auctioneer to see. In some protocol, we don't really need broadcast, we will specify where is the message from and sent to in those protocols.

2. When we use "**mechanism**," it means that players might not tell the truth and we need to incentivize the players to cooperate. A mechanism in this paper can be considered as a protocol together with an allocation rule and a payment rule. Let the protocol be π and the transcript be Π. For $i = 1, ..., n$, let S_i be the allocation rule and p_i be the payment of player i. Player i's utility is defined as $u_i(\Pi) = v_i(S_i(\Pi)) - p_i(\Pi)$.

Player i's goal is to maximize her expected utility $\mathrm{E}[u_i(\Pi)]$. The expectation is over the randomness of the mechanism.

We further define the truthful mechanism as the following. Let m_i be the message sent by player i. m_i is a function of v_i and the history of the protocol. Here we only make the definition for the case when each player sends at most one message in the protocol and all the mechanisms in this paper are in this case. We say that m_i is a **dominant strategy** (in expectation) for player i, if for all $v_1, ..., v_n$, player i's other strategy m'_i and other players' strategy m_{-i},

$$\begin{aligned} & \mathrm{E}\left[v_i(S_i(\Pi(m_i, m_{-i})) - p_i(\Pi(m_i, m_{-i}))\right] \\ \geq \; & \mathrm{E}\left[v_i(S_i(\Pi(m'_i, m_{-i})) - p_i(\Pi(m'_i, m_{-i}))\right]. \end{aligned}$$

We say that a mechanism is a **truthful mechanism** if there exist dominant strategies for all players.

One of our goals in this paper is to find an allocation that achieves good approximation of the maximum social welfare $\mathcal{SW}^*(v_1, ..., v_n)$ (defined as allocation problem in Section 4.1). We say a protocol is α-approximation if for all $v_1, ..., v_n$,

$$\mathrm{E}[\sum_{i=1}^{n} v_i(S_i(\Pi))] \geq \alpha \cdot \mathcal{SW}^*(v_1, ..., v_n).$$

We say a truthful mechanism is α-approximation if for all $v_1, ..., v_n$ there exist dominant strategies $m_1, ..., m_n$ for player $1, ..., n$ guaranteeing:

$$\mathrm{E}[\sum_{i=1}^{n} v_i(S_i(\Pi(m_1, ..., m_n))] \geq \alpha \cdot \mathcal{SW}^*(v_1, ..., v_n).$$

Below are definitions of the valuation classes used in the paper. These are equivalent to the definitions used in Section 4.1, but more apt for proofs and less apt for posing easy-to-parse communication problems.

Definition 4.1.3. *We consider the following classes of valuations:*

- *A valuation function v is **additive** if for every bundle S, $v(S) = \sum_{i \in S} v(\{i\})$.*
- *A valuation function v is **XOS** if there exist additive valuations $a_1, ..., a_t$ such that for every bundle S, $v(S) = \max_{i=1}^{t} a_i(S)$. Each a_i is called a **clause** of v.*
- *A valuation function v is **binary additive** if v is additive and for every item i, $v(\{i\}) \in \{0, 1\}$. We will sometimes refer to a binary additive valuation as a **set**, referring to $\{i | v(\{i\}) = 1\}$.*
- *A valuation function v is **binary XOS** if v is XOS and all v's clauses are binary additive valuations. Again, we will sometimes refer to v's clauses as **sets** to make it more natural to talk about unions/intersections/etc.*

4.1.5 Background on Related Work

There is an enormous literature of related work on combinatorial auctions. The state-of-the-art without concern for incentives is a 1/2-approximation for any number of subadditive bidders [97], and numerous improvements for special cases, such as submodular bidders [82, 97, 98]. With concern for incentives, the state-of-the-art (for worst-case approximation ratios and dominant strategy truthfulness) is an $1/O(\sqrt{\log m})$-approximation for XOS bidders, again with improvements for further special cases [86]. The problem has also been studied in Bayesian settings, where a generic black-box reduction is known if the designer only desires *Bayesian* truthfulness[11] [118, 117, 25]. If the designer desires dominant strategy truthfulness but is okay with an average-case welfare guarantee, then a 1/2-approximation is known for XOS bidders [101]. Combinatorial auctions have also been studied through the lens of *Price of Anarchy*, but a deeper discussion of this is outside the scope of this paper [27, 163, 180, 181, 100, 48, 78, 70, 140, 62, 38, 99].

[11] A mechanism is Bayesian truthful if it is in every bidder's interest to tell the truth, assuming all other bidders tell the truth and have values drawn from the correct Bayesian prior.

The direction of "truthful mechanisms versus algorithms" is also studied through other topics. For example, [165] introduces the *combinatorial public projects* problem, and characterize truthful mechanisms via a Roberts-like theorem [172]. They further show a separation between what is achievable by communication-efficient truthful mechanisms and communication-efficient algorithms, owing to this characterization. In contrast, such a characterization is not known (and not believed to exist) for combinatorial auctions, with Dobzinski's recent reduction being the only progress in this direction [77]. Nisan and Ronen's seminal paper also attacked this question through the problem of truthful job scheduling on unrelated machines [161]. Here, the specific question studied is fundamentally different: they ask whether or not any truthful mechanism (regardless of computation/communication) can achieve makespan guarantees competitive with the best possible (whereas for combinatorial auctions, the VCG mechanism guarantees that truthful mechanisms can achieve the first-best without concern for computation/communication [187, 63, 110]).

On the topic of simultaneous versus interactive communication, [190] proposed the 2-party simultaneous communication model when communication complexity was introduced. [166], [88], [162] showed that in the 2-party case, there is an exponential gap between k and $(k-1)$-round deterministic/randomized communication complexity of an explicit function. In the multiparty number-on-forehead communication model [51], [19] showed an exponential gap between simultaneous communication complexity and communication complexity for up to $(\log n)^{1-\varepsilon}$ players for any $\varepsilon > 0$. [79] recently showed that in combinatorial auctions with unit demand bidders/subadditive bidders, there is an exponential gap (exponential in the number of players) between simultaneous communication complexity and communication complexity. In comparison to these works, our separation between simultaneous and interactive communication for the 2-player BXOS decision problem is of a quite different flavor, and makes the available toolkit for future results more diverse.

4.2 Intuition for the Gap: an Extremely Toy Setting

Consider the following very toy setting: Alice and Bob each have some valuation function $v(\cdot)$ such that $v([m]) \in [1, M]$, and $v(\cdot)$ is monotone (no other assumptions).[12]

Observation 4.2.1. *In the very toy setting, Alice and Bob can guarantee the following tight approximation guarantees with zero communication:*

- *A* $1/2$*-approximation for the allocation problem with a randomized protocol: give all the items either to Alice or Bob uniformly at random.*
- *A* $1/(M+1)$*-approximation for the allocation problem with a deterministic protocol: give all the items to Alice.*
- *A* $1/(2M)$*-approximation for the decision problem (decide if social welfare* $\geq X$ *or* $\leq X/(2M)$*, arbitrary behavior allowed in-between): If* $X > 2M$ *output "*$\leq X/(2M)$*" If* $X \leq 2M$*, "*$\geq X$*."*

Since this example is just to provide intuition, we omit a complete proof. The first bullet should be fairly clear: the optimal welfare is clearly upper bounded by $v_1([m]) + v_2([m])$, and the protocol guarantees exactly half of this. The third bullet should also be clear: the optimal welfare is always between 1 and $2M$. Moreover, any value in the range is possible ($2M$ if, for instance, $v_1(\{1\}) = M = v_2(\{2\})$. 1 if, for instance, $v_1(S) = v_2(S) = 1$ iff $S \ni 1$, and $v_1(S) = v_2(S) = 0$ otherwise). So with zero communication, better than $1/(2M)$ is not possible. The middle bullet is perhaps the only tricky one. If we give all of the items to Alice, we guarantee welfare $v_1([m]) \geq 1$, and the optimum is upper bounded by $v_1([m]) + M$.

Again, the purpose of this example is just to provide intuition as to where this gap might come from, and we do not consider it a "result." Of course, one should not expect the gaps to stay quite so drastic as we dial up the communication: with just $\log M$ bits in the above example, a deterministic protocol for the allocation problem and decision problem can both

[12] If one wishes, one could further restrict attention to submodular, XOS, etc., but this section is just supposed to be a toy model to provide some intuition, and we will not belabor this point.

guarantee a 1/2-approximation (output $v([m])$). But this example still captures some of the intuition as to where the gap comes from.

4.3 Warmup: Beating a 1/2-Approximation

Before explaining our protocol, consider the following thought experiment: say instead Alice and Bob are asked to just report a single clause from their valuation. What clause should they choose and how well will this protocol solve the allocation/decision problem? It's not too hard to see that the best they can do is to just report the largest clause in their list (maximizes $b_i([m])$ over all clauses b_j), which will obtain just a 1/2-approximation for each problem. Now, what if they each report *two* clauses from their valuations, can they do something more clever? Well, they should certainly try to report clauses that are large, as this lets the other know which sets they value the most. But they should also try to report clauses that are different, as this allows for more flexibility in an allocation that both parties value highly. It's perhaps not obvious what the right tradeoff is between large/different (or even exactly what "different" should formally mean), but it turns out that a good approach is for Alice and Bob to each output the two clauses in their list with the largest union (i.e. output b_i, b_j maximizing $\mathcal{SW}^*(b_i, b_j)$). Subject to figuring out how to translate this information into solutions, a slight variant of this protocol guarantees a 2/3-approximation for the allocation problem, and a 3/5-approximation for the decision problem, and the proof is actually quite simple. Note below that Theorem 4.3.1 holds only for BXOS, whereas Theorem 4.3.2 holds for general XOS. We'll provide both proofs below first, followed by a brief discussion.

Protocol 4 Simultaneous randomized warmup protocol for 2-party combinatorial auctions with binary XOS valuations

1: Alice finds b_1, b_2, b_3 among clauses of her valuation v_1 such that b_1 maximizes $b_1([m])$ and b_2, b_3 maximize $\mathcal{SW}^*(b_2, b_3)$. Then she picks j uniformly at random from $\{1, 2, 3\}$ and sends b_j to the auctioneer.
2: For each item i, the auctioneer allocates it to Alice if $b_j(\{i\}) = 1$; otherwise allocate it to Bob.

Theorem 4.3.1. *Protocol 4 gives a 2/3-approximation to the 2-party BXOS allocation problem.*

Proof. First, we want to claim that if Alice sents b_j to the auctioneer, then the resulting welfare is at least $\mathcal{SW}^*(b_j, v_2)$. This is actually an instantiation of a claim we will want to reference later, so we'll state a more general form below:

Claim 4.3.1. *Let b_1 be a binary additive valuation and v_2 be a binary XOS valuation. Then the allocation that awards to Alice all items such that $b_1(\{i\}) = 1$ achieves welfare equal to $\mathcal{SW}^*(b_1, v_2)$.*

Proof. Let A denote the set of items for which $b_1(\{i\}) = 1$, and consider any other allocation $(B, \bar{B})$. We first reason that we can remove from B all items $\notin A$ without hurting $b_j(B) + v_2(\bar{B})$. This is trivial to see, as b_j has value 0 for all items $\notin A$. Next, we reason that we can add to B any item $\in A$ without hurting $b_j(B) + v_2(\bar{B})$. To see this, observe that we are certainly increasing $b_j(B)$ by 1 when we make this change, as b_j is just additive and $b_j(\{i\}) = 1$ for all $i \in A$. In addition, we can't possibly decrease $v_2(\bar{B})$ by more than 1, as all of the clauses in v_2 are binary additive (and therefore have value at most 1 for any item). So again, the total change is only positive. At the end of these changes, observe that we have now transitioned from $(B, \bar{B})$ to $(A, \bar{A})$ without losing any welfare, and therefore $(A, \bar{A})$ is indeed optimal. □

Claim 4.3.1 immediately lets us conclude that the expected welfare guaranteed by Protocol 4 is at least $\frac{1}{3} \cdot \sum_{j=1}^{3} \mathcal{SW}^*(b_j, v_2)$. Now, let S and T be the optimal allocation to achieve $\mathcal{SW}^*(v_1, v_2)$. Let a be the clause of v_1 such that $a(S) = v_1(S)$. Let a' be the clause of v_2 such that $a'(T) = v_2(T)$. So $\mathcal{SW}^*(v_1, v_2) = a(S) + a'(T)$. From the protocol, we know that $b_1([m]) \geq a([m]) \geq a(S)$. Moreover, if U and U' are the allocation that achieves $\mathcal{SW}^*(b_2, b_3)$, then we know that $b_2(U) + b_3(U') = \mathcal{SW}^*(b_2, b_3) \geq \mathcal{SW}^*(a, b_1) \geq a(S) + b_1(T)$ (by definition of b_2, b_3). In expectation, the social welfare we get in the protocol is at least:

$$\begin{aligned}
& \frac{1}{3} \cdot \sum_{j=1}^{3} \mathcal{SW}^*(b_j, v_2) \\
\geq\ & \frac{1}{3} \cdot (b_1(S) + a'(T) + b_2(U) + a'(U') + b_3(U') + a'(U)) \\
\geq\ & \frac{1}{3} \cdot (b_1(S) + a'(T) + a(S) + b_1(T) + a'([m])) \\
\geq\ & \frac{1}{3} \cdot (b_1([m]) + a(S) + 2a'(T)) \\
\geq\ & \frac{1}{3} \cdot (2a(S) + 2a'(T)) = \frac{2}{3} \cdot \mathcal{SW}^*(v_1, v_2).
\end{aligned}$$

□

Protocol 5 Simultaneous deterministic warmup protocol for 2-party combinatorial auctions with XOS valuations

---

1: Alice finds b_1, b_2, b_3 among clauses of her valuation v_1 such that b_1 maximizes $b_1([m])$ and b_2, b_3 maximize $\mathcal{SW}^*(b_2, b_3)$. Bob finds b_4, b_5, b_6 among clauses of his valuation v_2 such that b_4 maximizes $b_4([m])$ and b_5, b_6 maximize $\mathcal{SW}^*(b_5, b_6)$. Alice sends b_1, b_2, b_3 to the auctioneer and Bob sends b_4, b_5, b_6 to the auctioneer simultaneously.

2: **For allocation problem:** Auctioneer finds $j \in \{1, 2, 3\}, j' \in \{4, 5, 6\}$ that maximizes $\mathcal{SW}^*(b_j, b'_{j'})$ and allocate items according to it.

3: **For decision problem:** Let X be the parameter in the decision problem. Auctioneer finds $j \in \{1, 2, 3\}, j' \in \{4, 5, 6\}$ that maximizes $\mathcal{SW}^*(b_j, b'_{j'})$. If $\mathcal{SW}^*(b_j, b'_{j'}) \geq 3X/5$, say "yes" ($\mathcal{SW}^*(v_1, v_2) \geq X$). If $\mathcal{SW}^*(b_j, b'_{j'}) < 3X/5$, say "no".

Theorem 4.3.2. *Protocol 5 gives a $3/5$-approximation to the 2-party XOS allocation problem and the 2-party XOS decision problem.*[13]

Proof. Let S and T be the optimal allocation to achieve $\mathcal{SW}^*(v_1, v_2)$. Let a be the clause of v_1 such that $a(S) = v_1(S)$. Let a' be the clause of v_2 such that $a'(T) = v_2(T)$. So $\mathcal{SW}^*(v_1, v_2) = a(S) + a'(T)$. From the protocol, we know that $b_1([m]) \geq a([m]) \geq a(S)$ and $b_4([m]) \geq a'([m]) \geq a'(T)$. Let U and U' be the allocation to achieve $\mathcal{SW}^*(b_2, b_3)$. We know

[13]XOS allocation problem and XOS decision problem are the obvious extensions of BXOS allocation problem and BXOS decision problem for non-binary clauses.

that $b_2(U)+b_3(U') \geq a(S)+b_1(T)$. Let W and W' be the allocation to achieve $\mathcal{SW}^*(b_5,b_6)$. We know that $b_5(W)+b_6(W') \geq a'(T)+b_4(S)$. Then we have

$$
\begin{aligned}
& \mathcal{SW}^*(b_1,b_5)+\mathcal{SW}^*(b_1,b_6) \\
\geq\ & b_1(W')+b_5(W)+b_1(W)+b_6(W') \\
\geq\ & b_1([m])+b_5(W)+b_6(W') \geq a(S)+a'(T)+b_4(S).
\end{aligned}
$$

Similarly we have

$$
\mathcal{SW}^*(b_2,b_4)+\mathcal{SW}^*(b_3,b_4) \geq a(S)+a'(T)+b_1(T).
$$

The social welfare we get in the protocol is at least

$$
\begin{aligned}
& \mathcal{SW}^*(b_j,b'_{j'}) \\
\geq\ & \frac{1}{5}\cdot(\mathcal{SW}^*(b_1,b_4)+\mathcal{SW}^*(b_1,b_5)+ \\
& \mathcal{SW}^*(b_1,b_6)+\mathcal{SW}^*(b_2,b_4)+\mathcal{SW}^*(b_3,b_4)) \\
\geq\ & \frac{1}{5}\cdot(b_1(S)+b_4(T)+2a(S)+2a'(T)+b_4(S)+b_1(T)) \\
\geq\ & \frac{1}{5}\cdot(b_1([m])+b_4([m])+2a(S)+2a'(T)) \\
\geq\ & \frac{3}{5}(a(S)+a'(T)) = \frac{3}{5}\mathcal{SW}^*(v_1,v_2).
\end{aligned}
$$

From this, it is easy to check that Protocol 5 gives a 3/5-approximation to both the 2-party XOS allocation problem and the 2-party XOS decision problem. □

So now there are two remaining questions: first, how does one generalize the reasoning in Protocols 4 and 5 to multiple clauses? And second, why the heck is there a difference between their guarantees for the allocation and decision problem for binary XOS valuations? For the first question, we'll postpone the details to Section 4.4, but just note here that our full protocols indeed makes use of similar reasoning. For the second, observe that Claim 4.3.1 is somewhat magical: if Alice's valuation is binary additive, and Bob's is binary XOS, then it is possible to allocate the items optimally *without any input from Bob* (other than the knowledge that his valuation is indeed binary XOS). While it's not obvious that Claim 4.3.1 should necessarily be quite so helpful (given that we do, in fact, get input from Bob), this turns out to be the crucial difference between the allocation and decision problem. At a high level, there is necessarily some information lost between Alice's valuation and her message (ditto for Bob). The decision problem requires us to deal with both losses, but Claim 4.3.1 lets certain kinds of protocols only worry about the loss from Alice.

4.4 Developing Good Summaries

In this section, we define "summaries" in some specific forms for binary XOS valuations. They are the main ingredients in our protocols and mechanisms. At a high level, the summaries are trying to simultaneously maximize the size of the reported clauses, while also keeping on eye on reporting "different" clauses. One can interpret the negative term as a "regularizer" that achieves this goal. The total size of the reported clauses corresponds to term $\sum_{i=1}^{m} x_i$ and we encourage reporting "different" clauses by having the term $-\sum_{i=1}^{m} \alpha \cdot x_i^2$.

Definition 4.4.1 (Summaries of binary XOS valuations)**.** *For a binary XOS valuation v, define its (k, α)-summary $(b_1, ..., b_k)$ as* $\text{argmax}_{b_1,...,b_k \in \{a_1,...,a_t\}} \sum_{i=1}^{m} (x_i - \alpha \cdot x_i^2)$*, where $a_1, ..., a_t$ are the clauses of v and $x_i = \frac{b_1(\{i\})+\cdots+b_k(\{i\})}{k}$.*

Remark 4.4.1. *For the summaries defined above, there might be multiple $(b_1, ..., b_k)$'s maximize the term. When we use a (k, α)-summary in some protocol, we will use an arbitrary one. Additionally, note that our warm-up protocols from Section 4.3 ask Alice and Bob to output both their $(1, 1/2)$-summary and their $(2, 2/3)$-summary, see examples below.*

Example 4.4.1. *For a* $(1, 1/2)$*-summary of some binary XOS valuation* v*, we will find* b_1 *among clauses of* v *that maximizes*

$$\sum_{i=1}^{m} \left(b_1(\{i\}) - \frac{1}{2} \cdot (b_1(\{i\}))^2 \right) = \sum_{i=1}^{m} b_1(\{i\})/2 = \frac{1}{2} b_1([m]).$$

Example 4.4.2. *For a* $(2, 2/3)$*-summary of some binary XOS valuation* v*, we will find* b_1, b_2 *among clauses of* v *that maximize*

$$\begin{aligned} & \sum_{i=1}^{m} \left(\frac{b_1(\{i\}) + b_2(\{i\})}{2} - \frac{2}{3} \cdot \left(\frac{b_1(\{i\}) + b_2(\{i\})}{2} \right)^2 \right) \\ = \ & \frac{1}{3} \sum_{i=1}^{m} \left(b_1(\{i\}) + b_2(\{i\}) - b_1(\{i\}) b_2(\{i\}) \right) \\ = \ & \frac{1}{3} \mathcal{SW}^*(b_1, b_2). \end{aligned}$$

Proofs of some simple properties of these summaries, and an extension of the definition to non-binary XOS valuations can be found in the full version of [39]. Essentially, what the lemmas are stating is that for any set A, the summaries defined above do a "good enough" job capturing Alice's (/Bob's) value for A. Note that "good enough" doesn't mean "captures $v(A)$ within a constant factor," as this is impossible with a sketch [23]. "Good enough" simply means that the summary can be used inside a similar approach to Section 4.3.

Once summaries from Alice and Bob are in hand, there are a couple natural ways to "wrap up" the allocation/decision problem. We'll formally name these and refer to them in future protocols:

- **Alice-Only Allocation (randomized):** Pick a clause uniformly at random from Alice's summary, award to Alice items for which that clause values at 1, and the rest to Bob.

- **Best Known Allocation (deterministic):** If Alice reports clauses $a_1, \ldots, a_k$, and Bob reports clauses $b_1, \ldots, b_k$, find i, j maximizing $\mathcal{SW}^*(a_i, b_j)$. Allocate items according to the allocation that yields $\mathcal{SW}^*(a_i, b_j)$.

- **Best Known Decision(α, X) (deterministic):** If Alice reports clauses $a_1, \ldots, a_k$, and Bob reports clauses $b_1, \ldots, b_k$, find i, j maximizing $\mathcal{SW}^*(a_i, b_j)$. If $\mathcal{SW}^*(a_i, b_j) \geq \alpha X$ say "yes" (guess that $\mathcal{SW}^*(v_1, v_2) \geq X$). Otherwise, guess "no" (guess that $\mathcal{SW}^*(v_1, v_2) < \alpha X$).

4.4.1 Our Protocols and Mechanisms

In this section, we'll describe all protocols used to provide our positive results. All protocols involve Alice and Bob reporting a (k, α)-summary, and then using the Alice-Only or Best Known Allocation, or making the Best Known Decision. All proofs can be found in the full version of [39]. We make two remarks before proceeding:

1. All of the high-level intuition for why the protocols work is captured by the summaries. Many of the actual proofs are different, but at a high level everything comes down to the fact that this class of summaries selects "the right" clauses to report for welfare maximization.

2. Any protocol that eventually uses the Alice-Only Allocation doesn't require Alice to report her entire summary (she can just draw the random clause herself as in Protocol 4 (and have communication m for any choice of k). While we state the guarantees for such protocols for a fixed k, one can actually take $k \to \infty$ without increasing the communication at all.

Theorem 4.4.1. *The following protocols achieve guarantees:*

Alice's summary	*Bob's summary*	*Wrap-up*	*Approximation*	*Problem*	*Valuations*
$(k, 1/2)$	$\bot$	*Alice-Only*	$3/4 - 1/k$	*Allocation*	*BXOS*
$(k, 1/3)$	$(k, 1/3)$	*Best Known Allocation*	$23/32 - 1/k$	*Allocation*	*XOS*
$(k, 1/3)$	$(k, 1/3)$	*Best Known Decision*	$23/32 - 1/k$	*Decision*	*XOS*

Before continuing, we briefly remark the following:

- The 3/4-approximation guaranteed by the protocol in the first row is tight: [82] showed that randomized, interactive protocols require exponential communication to beat a 3/4-approximation.
- The second and third protocols also work for general XOS.
- It is still open whether it is possible to beat 23/32 with a deterministic protocol for the allocation problem, but 23/32 is optimal for any protocol using the Best Known Allocation after Alice and Bob each report a (k, α_i)-summary.

Additional applications of our summaries appear in the full version of this paper, including a 2-round protocol guaranteeing a 3/4-approximation for general XOS valuations, and our strictly truthful mechanism. The strictly truthful mechanism essentially visits bidders one at a time, asks for a $(k, 1/2)$-summary on the remaining items, awards them the "Alice-Only Allocation" for their reported summary, and charges payments to ensure strict truthfulness.

4.5 Lower Bounds

Finally, we overview our lower bound for the BXOS decision problem (which implies Theorem 4.1.2). We begin with some intuition: Alice and Bob will each get exponentially many clauses of size $m/2$. These sets will be random, but not uniformly random.[14] Instead, they

[14] If they were uniformly random, then Alice and Bob can guarantee $3m/4$ in expectation by just reporting a single arbitrary clause, because two uniformly random sets of size $m/2$ have union $3m/4$ in expectation.

are drawn in such a way that the union of two random clauses of Alice and Bob has size $(3/4 - 1/108)m$ in expectation. At this point, the optimal welfare is $(3/4 - 1/108)m$ if we don't further adjust their inputs. Finally, we modify the construction either by hiding or not hiding a_0 within Alice's input and b_0 within Bob's input such that $a_0 \cup b_0 = [m]$, in a matter so that these sets are indistinguishable from the rest. Therefore, the answer to the decision problem rests on whether or not Alice and Bob each have this hidden set, but they have no means by which to convey this information as this set looks indistinguishable from the rest. The proof of Theorem 4.5.1 below can be found in the full version of [39].

Theorem 4.5.1. *For any constant $\varepsilon > 0$, there exists a distribution over binary XOS valuations such that no simultaneous, randomized protocol with less than $e^{2Cm/9}$ communication can guarantee an α-approximation to the 2-party BXOS decision problem with probability larger than $\frac{1}{2} + 2e^{-Cm/9}$. Here $\alpha = 3/4 - 1/108 + \varepsilon$ and $C = 2\varepsilon^2$.*

4.6 Discussion and Future Work

Our main result shows a simultaneous protocol guaranteeing a 3/4-approximation for the BXOS allocation problem, and a lower bound of $3/4 - 1/108$ for for the BXOS decision problem. The bigger picture behind these results, even without consideration of truthful combinatorial auctions, is the following:

- It is surprising that the decision problem is strictly *harder* than the allocation/search problem. To the best of our knowledge, this is the first instance of such a separation.
- It is surprising that a $(> 1/2)$-approximation for either the allocation or decision problem is possible at all, given the strong lower bounds already known on sketching valuation functions, but we are able to get a tight 3/4-approximation for the allocation problem.

- A 3/4-approximation for the decision problem now serves as a new example of what can be achieved in polynomial interactive communication (in fact, two rounds by a theorem in the full version), but requires exponential simultaneous communication. While such problems are already known, this has a very different flavor than previous constructions, and will likely be a useful tool for this reason.

The most obvious question is to resolve whether or not there is a 3/4-approximation for the allocation problem with general XOS functions. If there isn't, this would provide the first separation between truthful and non-truthful protocols with polynomial communication via Dobzinski's reduction [77]. Additionally, whether or not our protocol can be de-randomized is an enticing open question: if no matching deterministic protocol can be found (implying a lower bound of $< 3/4$ for deterministic protocols for the allocation problem), this would provide the first separation between truthful and non-truthful deterministic protocols (Dobzinski's reduction preserves determinism). If our protocol can in fact be de-randomized, this would be fascinating, as this protocol would *deterministically* guarantee a 3/4-approximation without learning the welfare it achieves.[15]

Finally, while we have provided simultaneous protocols for the allocation problem with approximation guarantees strictly better than 1/2 when bidders have XOS valuations, it still remains open whether or not a truthful mechanism can obtain a ($> 1/2$)-approximation for two-player combinatorial auctions with XOS bidders.

[15]It somehow seems tempting to conjecture both that our protocol can be de-randomized and that it can't - a random clause of Alice's does well on average with no input from Bob, so to de-randomize we just need Bob to tell us *something* that identifies a clause performing better than average. At the same time it seems extremely unlikely that a deterministic protocol will somehow provide an approximation guarantee better than $3/4 - 1/108$ for the allocation problem without violating Theorem 4.5.1.

Chapter 5

Interpolating Between Truthful and non-Truthful Mechanisms for Combinatorial Auctions

The results of this chapter are based on joint work with Mark Braverman and Matt Weinberg [38].

5.1 Introduction

In a combinatorial auction, a single designer has m items available for purchase to n buyers. Each buyer has some private valuation function $v_i(\cdot) : 2^{[m]} \to \mathbb{R}^+$ over subsets of items, and the seller aims to partition the items into $S_1 \sqcup \ldots \sqcup S_n$ so as to optimize the social welfare, $\sum_i v_i(S_i)$. Much recent work addresses the design of combinatorial auctions, targeting the desiderata of optimality, simplicity (from both a design and strategic perspective), and computational tractability. For instance, the celebrated Vickrey-Clarke-Groves mechansim achieves the optimal social welfare, and is *truthful* (therefore it is strategically simple: no bidder need consider any strategy except for honest behavior) [187, 63, 110]. However, in virtually all settings of interest, the VCG mechanism is not computationally tractable, making it unusable in practice.

Much recent work of computer scientists has targeted the design of auctions that are instead approximately optimal, but computationally tractable. One active line of work searches for truthful mechanisms [139, 80, 81, 74, 86, 134]. While these results all achieve computational tractability and strategic simplicity in the strongest possible way, the mechanisms are quite involved and therefore don't achieve design simplicity. More importantly, many of these mechanisms can only guarantee approximation ratios that are polynomial in m. When buyers are assumed to be submodular[1] or subadditive,[2] the best achieve ratios just logarithmic in m. A central open problem is the design of computationally tractable truthful mechanisms that guarantee a constant-factor approximation when valuation functions are submodular or subadditive.

Another exciting line of work has shown simple mechanisms that achieve a low *price of anarchy* at various equilibrium concepts [27, 163, 180, 181, 100]. These results show, for instance, that as long as buyers are subadditive and interact at equilibrium, auctioning each item simultaneously via a first-price auction achieves half the optimal social welfare [100]. All of these auctions are computationally tractable and simple in design, and many achieve approximation ratios that are very small constants, via the price of anarchy. However, *none* of the equilibria at which these results hold are known to arise naturally, and some are even known to be computationally intractable [48, 78].[3] Note that even distributed regret minimization may be computationally intractable in these settings, as each buyer has exponentially many (in m) strategies to consider. Therefore, these mechanisms are all extremely complex from a strategic perspective, as buyers would have to reason about

[1]A valuation function $v(\cdot)$ is submodular if $v(X) + v(Y) \geq v(X \cap Y) + v(X \cup Y)$ for all X, Y.

[2]A valuation function $v(\cdot)$ is subadditive if $v(X) + v(Y) \geq v(X \cap Y)$ for all X, Y.

[3]Some equilibrium concepts, such as a pure Nash equilibrium in simultaneous second price auctions for submodular buyers, can be found in polynomial time [140, 62, 78]. However, the algorithms finding them are highly centralized, and the equilibria themselves are very unnatural: each item only has only one non-zero bidder, even though bidding zero on any item is possibly a dominated strategy.

exponentially many different strategies in order to approach an equilibrium at which good approximation guarantees hold.

So, truthful mechanisms are strategically simple but achieve poor approximation ratios, and simple mechanisms achieve good approximation guarantees but are strategically complex. As an alternative to pursuing each direction separately, we propose taking ideas from each and introduce *interpolation mechanisms*. An interpolation mechanism has two phases. In the first phase, buyers participate in some non-truthful mechanism whose output is itself a truthful mechanism. In the second phase, buyers participate in the truthful mechanism selected during phase one. In this language, all truthful mechanisms are interpolation mechanisms with a non-existent first phase, and the simple mechanisms referenced above are interpolation mechanisms with a non-existent second phase.

What might interpolation mechanisms bring to the table that truthful mechanisms and existing simple mechanisms don't? This question is best addressed with an example. Recent work of Devanur et. al. [70] designs the first interpolation mechanism (although they did not consider this classification), the single-bid mechanism. Phase one of the single-bid mechanism asks each buyer to report just a single real number, b_i, as their bid. Phase two visits the buyers one by one in decreasing order of b_i, and allows the buyer to purchase any number of remaining items at b_i per item (so more items are available to higher bidders, but lower bidders pay less per item). It is easy to see that once the bids are fixed and order determined in phase one, phase two constitutes a truthful mechanism. Note that phase one by itself is extremely limited: buyers are asked to represent their entire valuation function (of which there are doubly-exponentially many) with just $\log m$ bits. Unsurprisingly, no protocol using this limited amount of communication can possibly find a good allocation directly. Note also that phase two by itself is also quite limited: an ordering of the bidders along with a single price is set ahead of time, then buyers do as they please. Also unsurprisingly, such truthful mechanisms (that we call *single-price* mechanisms) can't guarantee any non-trivial approximation ratio. From our perspective, the single-bid mechanism is interesting because it takes two useless mechanisms, neither of which can guarantee a sub-polynomial approximation ratio on even 0/1-additive buyers,[4] and combines them into a mechanism with a price of anarchy $O(\log m)$ at correlated equilibria when buyers are subadditive. Importantly, because the per-bidder communication in phase one is only logarithmic, each bidder can actually implement any standard regret minimization algorithm over possible bids in poly time. Therefore, the mechanism achieves design simplicity, strategic simplicity, and computational tractability. The main open problem left following their work is the design of mechanisms that achieve these three desiderata with a constant price of anarchy.

Interpolation mechanisms are a natural avenue to tackle this problem, and therefore lower bounds on their capability are important to guide their research. Following Devanur et. al.'s work, questions arose such as: what if bidders make a constant number of bids instead of just one? What if the posted prices are different for each item? What if we restrict attention to a much smaller class than subadditive bidders? What if we consider price of stability instead of anarchy? Surprisingly, a subset of our results shows that *none* of these relaxations suffice to (significantly) beat the $O(\log m)$ bound attained by the single-bid mechanism. The remainder of our results show that lower bounds known for various classes of truthful mechanisms also extend to interpolation mechanisms with little first-phase communication. One should interpret these results *not* as claiming that the limits of interpolation mechanisms have already been reached, but as guiding future research towards other classes of truthful mechanisms (specifically, we identify posted-price mechanisms as a natural candidate in Section 5.1.2).

5.1.1 Our Results

In addition to formally identifying interpolation mechanisms as an important avenue of study, we identify their connection to the price of anarchy and stability, and provide numerous lower bounds. Our lower bounds consider interpolation mechanisms where the phase-two

[4] A buyer is additive if they have a value v_i for item i, and their value for a set S is $\sum_i v_i$. A buyer is 0/1-additive if each $v_i \in \{0, 1\}$.

mechanism comes from a certain class. The goal of these lower bounds is to identify which classes of mechanisms are incompatible with interpolation (MIR and value-query, below), and for which classes of mechanisms the limits have already been reached (single-price and non-adaptive posted-price, below) to guide future research towards others (adaptive posted-price mechanisms, Section 5.1.2). Our results all lower bound the amount of first-phase communication necessary to find a suitable phase-two mechanism from the desired class. Note that even for truthful mechanisms, no unconditional communication lower bounds are known outside of artificial settings, so it is outside the scope of this paper to suddenly provide unconditional lower bounds in the strictly more general setting of interpolation mechanisms.

Price of Anarchy and Price of Stability. Any bidder participating in an interpolation mechanism with $O(\log m)$ first-phase communication per bidder can run any standard regret minimization algorithm in poly-time. Because bidders need not strategize over their phase-two behavior, they need only optimize over their possible strategies in phase one, of which there are at most poly(m). Therefore, price of anarchy bounds for correlated equilibria of interpolation mechanisms with logarithmic first-phase communication have some extra bite, as bidders can be reasonably expected to converge to a correlated equilibria and the bound will hold. We call such mechanisms *a priori learnable*, and formally define this in Section 5.2.

Because we lower bound the first-phase communication complexity, we not only lower bound the achievable price of anarchy by such mechanisms, but also the price of stability. In this context, our bounds are strong in the sense that they don't rely on equilibrium behavior of the buyers, and apply no matter how the buyers interact. Prior to this work, Roughgarden provides the only general approach for proving price of anarchy lower bounds [174], and no general approach was known for price of stability at all. Our approach is similar to Roughgarden's in the sense that both identify settings in which communication lower bounds imply "the right" price of anarchy lower bounds. Still, our approach differs signifcantly as Roughgarden's work specifically targets equilibrium concepts that are *not* efficiently computable, and doesn't apply to price of stability. We discuss formally the connection between first-phase communication bounds and price of anarchy/stability in Section 5.2.

Single-Price Mechanisms. A single-price mechanism fixes a price p_i for buyer i, then visits the buyers one at a time and offers buyer i any remaining items for p_i each. Devanur et. al.'s single-bid mechanism has $O(\log m)$ first-phase communication per bidder, and obtains a price of anarchy at correlated equilibria of $O(\log m)$ whenever buyers are subadditive. We show in Section 5.3 that even when buyers are just additive, *no amount of first-phase communication* suffices for an interpolation mechanism whose second phase is a single-price mechanism to obtain an approximation ratio $o(\log m/ \log\log m)$. Note that this *significantly* improves a lower bound shown in [70], which simply proved that the single-bid mechanism itself could not guarantee an approximation ratio $o(\log m/ \log\log m)$.

Non-Adaptive Posted-Price Mechanisms. Non-adaptive posted-price mechanisms generalize single-price mechanisms by allowing the mechanism to set a price p_{ij} for buyer i to purchase item j. The mechanism still visits the buyers one at a time, and allows buyer i to purchase any remaining items at the designated price. We show in Section 5.4 that even when buyers are just additive, any interpolation mechanism whose second phase is a non-adaptive posted-price mechanism and guarantees an $o(\log m/ \log\log m)$ approximation ratio has $\Omega(m^{1-\epsilon})$ first-phase communication per bidder, for all $\epsilon > 0$. Therefore, the single-bid mechanism cannot be improved by restricting attention to a smaller class of valuations, restricting attention to a smaller class of equilibrium concepts, setting different prices for different items, or allowing significantly more (but still sublinear) first-phase communication.

Maximal-In-Range, Value Query, and Computationally Efficient Mechanisms. Several recent works have identified lower bounds on approximation ratios that can possibly be obtained by these classes of mechanisms, which we will define in the corresponding sections. We extend these lower bounds to mechanisms with low first-phase communication that induce a mechanism in one of these classes. In Section 5.5, we extend techniques of Daniely et. al. based on generalizations of the VC-dimension [67], and in Section 5.6, we extend the techniques of Dobzinski and Vondrak based on structured sub-menus [75, 83].

5.1.2 Discussion and Future Work

Motivated by impossibility results associated with truthful mechanisms, and concerns regarding the strategic simplicity of existing simple mechanisms analyzed via price of anarchy, we propose the study of interpolation mechanisms. Using this new notion, we show that the single-bid mechanism of Devanur et. al. [70] is essentially optimal for its class, even subject to quite significant generalizations. We note that, prior to our work, it was unclear even how to define a class containing this mechanism, let alone prove lower bounds against mechanisms "like this." We also identify several classes of truthful mechanisms that are incompatible with interpolation in the sense that low first-phase communication doesn't allow for better approximation guarantees than no first-phase communication.

Our work identifies *adaptive* posted-price mechanisms (where the mechanism may choose what prices to set based on what items have already sold) as an intriguing class of mechanisms to study with interpolation, as none of the lower bounds from this work apply. Furthermore, Dynkin's secretary algorithm [94] immediately implies an adaptive posted-price mechanism that gets a $1/e$ approximation for additive bidders, so mild adaptations of our lower bounds for non-adaptive posted-price mechanisms are unlikely to apply. Can an interpolation mechanism with $O(\log m)$ per bidder first-phase communication and an adaptive posted-price mechanism for its second phase guarantee a constant price of anarchy?

Our results also fit into a line of work designing combinatorial auctions with low price of anarchy via *valuation compression* [89, 120, 90, 21]. These mechanisms restrict the allowable valuation reports from buyers to a space where the VCG mechanism is computationally tractable, even though the buyers may have much more complex valuations. In our context, these mechanisms still consist of just a first phase, and therefore rich valuation classes (like submodular, subadditive, or even just additive) cannot be compressed all the way down to a class that can be indexed with just $O(\log m)$ bits without super-constant loss. On this front, interpolation mechanisms provide a new style of two-phase valuation compression where this level of compression may be attainable.

Additionally, many existing truthful auction formats are naturally parameterized by parameters that are assumed to be known to the designer (e.g. buyers' budgets in a clinching auction [106, 107, 108] or buyers' interest sets in single-minded combinatorial auctions). Our framework provides a natural extension of such mechanisms to settings where these parameters are instead private. For instance, one could take any clinching auction where the budgets are assumed to be known, and add a first phase where buyers are asked to report their private budget. It would be very interesting to analyze the price of anarchy of such interpolation mechanisms, as these parameters are often not public knowledge in practice.

Finally, while we were motivated to study interpolation mechanisms for welfare maximization in combinatorial auctions, interpolation will also be useful in any setting where unfortunate lower bounds are known for truthful mechanisms but strategic simplicity is still a concern. A natural generalization of the presented setting, which we omit due to space constraints, is a model where rounds of truthful and non-truthful interaction might be interleaved (instead of having all non-truthful interaction come before all truthful interaction). It would be interesting to understand the power and complexity of such mechanisms in settings beyond necessarily just combinatorial auctions.

5.2 Preliminaries

In a combinatorial auction, the designer has m items to allocate to n buyers. Each item can be allocated to at most one buyer, and the buyers can be charged any non-negative price. Agents have a valuation function $v_i(\cdot)$ mapping subsets of items to non-negative real values.

Agents are quasi-linear, meaning that their utility for receiving items S_i and paying price p_i is $v_i(S_i) - p_i$. The designer's goal is to select an allocation that (approximately) maximizes the *welfare*, $\sum_i v_i(S_i)$.

A mechanism is *truthful* if it is in every buyer's interest to tell the truth, no matter their type. Formally, if $p_i(\vec{v})$ denotes the expected price paid by buyer i when the reported types are $\vec{v}$, and $S_i(\vec{v})$ denotes the (possibly random) set that buyer i receives, then we must have:

$$\mathbb{E}_{S_i \leftarrow S_i(\vec{v})}[v_i(S_i)] - p_i(\vec{v})$$

$$\geq \mathbb{E}_{S_i \leftarrow S_i(\vec{v}_{-i};v_i')}[v_i(S_i)] - p_i(\vec{v}_{-i}; v_i'), \quad \forall i, \vec{v}_{-i}, v_i, v_i'.$$

We define various classes of mechanisms and subclasses of valuation functions within the following sections.

5.2.1 Interpolation Mechanisms

An interpolation mechanism is a communication protocol with two phases. The first phase is non-truthful, and the output is a truthful mechanism. The second phase is the truthful mechanism output in phase one, and the output is an allocation of items and prices to charge.

Definition 5.2.1. *(Interpolation Mechanism) Let $\mathcal{M}$ denote the space of all truthful mechanisms for a combinatorial auction setting. Note that the output space of all $M \in \mathcal{M}$ is an allocation of items and charged prices. An interpolation mechanism provides a communication protocol, P, that outputs a mechanism $M \in \mathcal{M}$ based on the transcript of P. In phase one, bidders participate in the protocol P. In phase two, bidders participate in the truthful mechanism output by P during phase one. After phase two, the items are allocated and prices charged according to the bidders' play of the phase two mechanism. If the second phase of an interpolation mechanism always lies inside a restricted class $\mathcal{C}$ of truthful mechanisms, then we call this a "$\mathcal{C}$ interpolation mechanism."*

Our main results provide lower bounds on the per-bidder communication necessary during the first phase in order to possibly select a good truthful mechanism for the second phase. Formally, we say that an interpolation mechanism guarantees an approximation ratio of c when buyers have types in $\mathcal{V}$ if for all $i, v_i \in \mathcal{V}$, there exists a phase-one strategy for buyer i, $s_i(v_i)$, such that for all $\vec{v} \in \mathcal{V}^n$, if buyers use the strategies $s_i(v_i)$ during phase one, and report truthfully during phase two, the resulting allocation obtains a $1/c$-fraction (in expectation) of the optimal social welfare for $\vec{v}$.

Note that this approximation guarantee is not tied to any particular equilibrium concept. It is strictly easier to design an interpolation mechanism that guarantees an approximation ratio of c than one that has a price of anarchy/stability of c (stated formally in the following section), so lower bounds on the approximation ratio imply lower bounds on attainable price of anarchy/stability.

Of specific interest are interpolation mechanisms that have $\text{poly}(n, m)$ total communication, and only require bidders to consider $\text{poly}(m)$ strategies. Note that bidders must, at least a priori, consider every possible strategy during phase one (but need only consider telling the truth during phase two). So in order to guarantee that bidders have at most $\text{poly}(m)$ strategies to consider, the first phase must be especially simple.

Definition 5.2.2. *(a priori learnable) We say that an interpolation mechanism is **a priori learnable** if the first phase contains a single simultaneous broadcast round of communication, and the per-bidder communication is* $O(\log m)$.[5]

Observation 5.2.1. *Any buyer can run any standard regret minimization algorithm (for instance, Multiplicative Weights Updates) over her strategies in an a priori learnable interpolation mechanism in time/space* $\text{poly}(m)$. *Therefore, a correlated equilibrium of any a priori learnable interpolation mechanism can be found in poly-time, and correlated equilibria arise as the result of poly-time distributed regret minimization.*

[5]Note that, for instance, a single simultaneous broadcast round of $\text{poly}(m)$ communication per bidder results in exponentially many strategies (as in simultaneous first or second price auctions).

Proof. As the second phase is a truthful mechanism, each buyer need not strategize over possible actions during the second phase. Therefore, buyers should always play their dominant strategies during phase two and need only learn over their strategies during phase one; this can only decrease their regret. As phase one is a normal form game and there are only poly(m) such strategies, each buyer can just run a standard regret minimization algorithm in time/space poly(m). If each buyer does this, their play will converge to a correlated equilibrium [103, 115]. □

Note that price of anarchy bounds for correlated equilibria in a priori learnable interpolation mechanisms have more bite than price of anarchy bounds for solution concepts that don't arise naturally. The single-bid mechanism, for instance, is a priori learnable.

5.2.2 Connection to Price of Anarchy and Stability

The main application of our first-round communication lower bounds is on the price of anarchy or stability achievable for any a priori learnable interpolation mechanism. Price of anarchy/stability is typically defined for the social welfare, but has recently been considered also for revenue [116], and is well-defined for more general objectives as well. The observation below holds for any objective, but we state it for social welfare in combinatorial auctions since that is the focus of this paper.

Definition 5.2.3. *Let E denote any solution concept (i.e. Nash equilibria) for the mechanism M, and $\mathcal{V}$ denote any set of valuation functions. Then the price of anarchy (PoA) and Price of Stability (PoS) of M with respect to E when buyers have valuations in $\mathcal{V}$ are:*

$$PoA = \max_{\vec{v} \in \mathcal{V}^n} \frac{\max_{S_1 \sqcup \ldots \sqcup S_n} \{\sum_i v_i(S_i)\}}{\min_{\vec{s} \in E} \{\mathbb{E}_{S_1,\ldots,S_m \leftarrow M(\vec{s})}[\sum_i v_i(S_i)]\}}.$$

$$PoS = \max_{\vec{v} \in \mathcal{V}^n} \frac{\max_{S_1 \sqcup \ldots \sqcup S_n} \{\sum_i v_i(S_i)\}}{\max_{\vec{s} \in E} \{\mathbb{E}_{S_1,\ldots,S_m \leftarrow M(\vec{s})}[\sum_i v_i(S_i)]\}}.$$

Observation 5.2.2. *If an interpolation mechanism has price of anarchy or price of stability α at any non-empty equilibrium concept, then that same interpolation mechanism guarantees an approximation ratio of α. Therefore, lower bounds on the approximation ratios of interpolation mechanisms imply lower bounds on the possible price of anarchy/stability obtainable by those same mechanisms.*

Proof. Just sample each strategy $s_i(v_i)$ from any equilibrium where the price of anarchy/stability holds. This strategy immediately witnesses that the interpolation mechanism guarantees an α-approximation. □

5.3 Single-Price Mechanisms

In this section, we consider *single-price mechanisms.* A single-price mechanism visits bidders one at a time and offers the current bidder the opportunity to buy any number of remaining items at p_i per item. The main result of this section is the following:

Theorem 5.3.1. *There exist profiles of additive buyers for which the best single-price mechanism achieves an $\Omega(\log m/\log\log m)$-approximation. Therefore, for all $C > 0$, no single-price interpolation mechanism with first-round communication C per bidder obtains an $o(\log m/\log\log m)$-approximation on all profiles of additive buyers. This holds even when each buyer values each item at an integer between 1 and m.*

Proof. Consider the following example. There are b buckets of items (indexed from 0 to $b-1$), with bucket i containing c^{b-i} items, for some constants b, c to be set later. The value of (almost) every bidder for each item in bucket i is c^i. Each item is "special" for exactly one bidder, who values it instead at c^{i+1}. Each bidder has exactly c^{b-i}/n special items in bucket i. It is clear that the optimal allocation in this instance is to award each bidder each of their special items, which has welfare bc^{b+1}.

Now consider any single-price mechanism, with prices $p_1, \dots, p_n$. We want to consider when bidder i will get his special items in bucket j. Notice that bidder i's special items in bucket j are available to her if and only if $p_k > c^j$ for all $k < i$. Bidder i will choose to purchase her special items in bucket j if and only if $p_i \leq c^{j+1}$.

So for each bucket j, let i_j denote the first bidder for which $p_i \leq c^j$ (w.l.o.g. such a bidder exists as it is always optimal to set $p_n = 0$), and n_j denote the number of bidders before i whose price is at most c^{j+1}. Then the number of bidders who get their special items in bucket j is exactly $n_j + 1$. So the total number of pairs (i, j) such that bidder i gets her special items in bucket j is exactly $b + \sum_{j=1}^{b} n_j$. It's also clear that $\sum_{j=1}^{b} n_j \leq n$, as $p_i \in (c^j, c^{j+1}]$ for at most one j. So the number of pairs (i, j) such that bidder i gets her special items in bucket j is at most $b + n$.

Finally, observe that if the number of pairs (i, j) such that bidder i receives her special items in bucket j is x, then the welfare is exactly $xc^{b+1}/n + (bn - x)c^b/n$, which achieves at most a $(\frac{x}{nb} + \frac{1}{c})$-fraction of the optimal welfare. Plugging in for $x = n + b$, this is a $1/(1/n + 1/b + 1/c)$-approximation.

Setting $b = c = n$ provides an example with $m = \Theta(n^n)$ items (so $n = \Theta(\log m / \log\log m)$) for which no single-price mechanism obtains an $o(n) = o(\log m / \log\log m)$-approximation.

□

Notice that the impossibility above is quite strong: no amount of communication suffices to find a good single-price mechanism (because it is possible that one simply doesn't exist). This greatly strengthens an inapproximability result of [70], which just shows that their specific procedure (the single-bid mechanism) for selecting one doesn't obtain a better approximation ratio.

Corollary 5.3.1. *No single-price interpolation mechanism obtains a price of anarchy or price of stability $o(\log m / \log\log m)$ at any solution concept that is guaranteed to exist on all profiles of additive buyers.*

5.4 Non-Adaptive Pricing Mechanisms

In this section, we consider *non-adaptive posted-price* mechanisms. A non-adaptive posted-price mechanism orders the bidders however it wants (possibly randomly), then selects a price vector $\vec{p_i}$ for each bidder i. The bidders are visited one at a time, and offered the opportunity to purchase any subset S_i of remaining items for price $\sum_{j \in S_i} p_{ij}$. The main result of this section is below. Our proof uses the probabilistic method, which has also been used in [91] to prove price of anarchy lower bounds.

Theorem 5.4.1. *Any non-adaptive posted-price interpolation mechanism that guarantees an approximation ratio of $o(\log m / \log \log m)$ on all profiles of additive bidders has first-round communication at least $m^{1-\epsilon}$ per bidder, for all $\epsilon > 0$. This holds even when each buyer values each item at an integer between 1 and m.*

Proof. We will use the probabilistic method to define a set of profiles of additive bidders such that no non-adaptive posted-price mechanism does well on much of the set. Let each $\vec{v}_j$ (the vector of values of each bidder for item j) be drawn independently, and be equal to a random permutation of $(c^{k+1}, c^k, \ldots, c^k)$ with probability $1/c^k$ for each $k \in \{1, \ldots, b\}$, and $(0, \ldots, 0)$ with probability $1 - \sum_{k=1}^{b} 1/c^k$ for constants $c \geq 2, b$ to be set later.

It is clear that the expected maximum value per item is exactly bc, so the expected optimal welfare is bcm. Consider now any non-adaptive posted-price mechanism, and restrict attention to prices for item j. For each k, let i_k denote the first bidder such that $p_{i_k j} \leq c^k$, and n_k denote the number of bidders before i_k such that $p_{ij} \leq c^{k+1}$. Then the probability that this mechanism awards the item to the "special" bidder when the profile is a random permutation of $(c^{k+1}, c^k, \ldots, c^k)$ is exactly $\frac{1+n_k}{n}$. Therefore, the expected welfare of this posted-price mechanism, just considering contributions from item j, is $\sum_{k=1}^{b} c(1+n_k)/n + (n-1-n_k)/n$. It is also clear that $\sum_{k=1}^{b} n_k \leq n$, as each $p_{ij} \in (c^k, c^{k+1}]$ for at most one k. So the expected welfare per item of this non-adaptive posted-price mechanism is at most $cb/n + c + b$, and the total expected welfare is at most $(cb/n + c + b)m$.

Because the values for each item are drawn independently, the optimal welfare and the welfare of this non-adaptive posted-price mechanism is the sum of m independent random variables, each in $[0, c^{b+1}]$. Therefore, we can use the Chernoff bound to bound the probability that these random variables deviate from their expectation.

Set $b = c = n$. Then the probability that the welfare of any fixed item pricing exceeds $2(3n)m$ is at most e^{-m/n^n}. The probability that the optimal welfare is less than $(n^2m)/2$ is at most $e^{-m/(4n^{n-1})}$. So consider any set P of at most $2^{m/n^n}$ different non-adaptive posted-price mechanisms. Taking a union bound over all mechanisms $M \in P$, we see that with non-zero probability, the welfare of M is at most $6nm$ while the optimal welfare is at least $n^2m/2$. Therefore, there exists a profile of additive bidders for which no mechanism in P is an $n/12$-approximation.

If the first-round communication of each player is at most m/n^{n+1}, then there are only $2^{m/n^n}$ possible transcripts from the first round, and therefore only $2^{m/n^n}$ different non-adaptive posted-price mechanisms can possibly result. By the above reasoning, this implies the existence of a profile for which every possible mechanism selected (and therefore every outcome selected by the protocol) does not obtain an $n/12$-approximation. For any fixed ϵ, setting $m = n^{n/\epsilon}$ yields an instance with $n = \Theta(\epsilon \log m / \log \log m)$ that requires $m^{1-\epsilon}$ first-round bits per bidder to optain an $n/12$ approximation. □

Interestingly, there is always a non-adaptive posted-price mechanism that allocates the items optimally: set $p_{ij} = \max_{i' \neq i} v_{i'j}$ for all i, j. Each $v_{i'j}$ can be communicated with $\log m$ bits, so the entire mechanism can be found with $m \log m$ bits of communication per bidder. The theorem states that sublinear communication doesn't suffice to find a very good mechanism.

Corollary 5.4.1. *No a priori learnable non-adaptive posted-price interpolation mechanism obtains a price of anarchy or price of stability $o(\log m/ \log\log m)$ at any solution concept that is guaranteed to exist on all profiles of additive buyers.*

5.5 Maximal-In-Range Mechanisms

In this section, we consider *maximal-in-range* (MIR) mechanisms. A maximal-in-range mechanism selects some subset $\mathcal{F}' \subseteq 2^{[n]\times[m]}$ of feasible allocations, and *always* selects an outcome in $\arg\max_{x\in\mathcal{F}'}\{\textsc{Welfare}(x)\}$ (where the welfare is computed with respect to the valuation profile). In other words, a maximal-in-range mechanism always optimizes welfare exactly over a restricted set of possible outcomes. We provide a mild generalization of the techniques of Daniely et. al. [67] that apply to MIR interpolation mechanisms rather than just MIR mechanisms. With these new techniques, we show the following theorem. All proof details can be found in the full version of [38].

Theorem 5.5.1. *For all $\delta > 0$, the following hold:*

- *Assuming $NP \subsetneq P/poly$, any poly-time (runs in time $\text{poly}(n, m)$) MIR interpolation mechanism that obtains an approximation ratio $m^{1/3-2\delta/3}$ whenever buyers are single-minded*[6] *has first-round communication at least m^δ per bidder.*
- *Assuming $NP \subsetneq P/poly$, any poly-time (runs in time $\text{poly}(n, m)$) MIR interpolation mechanism that obtains an approximation ratio $m^{1/3-\delta}/5$ whenever buyers are capped-additive*[7] *has first-round communication at least $m^{1/3}$ per bidder.*
- *Any poly-communication (total communication $\text{poly}(n, m)$) MIR interpolation mechanism that obtains an approximation ratio $m^{1/3-\delta}$ whenever buyers are submodular*[8] *has first-round communication at least $m^{1/3}$ per bidder.*

Corollary 5.5.1. *Assuming $NP \subsetneq P/poly$, no a priori learnable, computationally efficient MIR interpolation mechanism obtains a price of anarchy or price of stability $o(m^{1/3})$ at any solution concept that is guaranteed to exist on all profiles of single-minded buyers, capped-additive buyers, or submodular buyers.*

[6] A valuation function $v(\cdot)$ is single-minded if there is a special set S and $v(T) = v(S)$ for all $S \subseteq T$, and $v_i(T) = 0$ otherwise.

[7] A valuation function $v(\cdot)$ is capped-additive if there is a budget b such that $v(S) = \min\{b, \sum_{j\in S} v(\{j\})\}$.

[8] A valuation function $v(\cdot)$ is submodular if $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$ for all S, T.

5.6 Value Query and Computationally Efficient Mechanisms

In this section, we consider value query mechanisms and arbitrary computationally efficient mechanisms. A mechanism is a *value query* mechanism if it only interacts with buyer valuations with queries of the form: "what is your value for set S?" A computationally efficient mechanism is any mechanism that terminates in polynomial time in m, n, and the space it takes to describe a valuation function. Note that for single-minded and capped-additive functions, the space required is also poly(m, n), but for submodular functions the space required may be larger. We provide a mild generalization of techniques of Dobzinski and Vondrak [75, 83] that apply to interpolation mechanisms rather than just truthful mechanisms. With these new techniques, we show the following theorem. All proof details can be found in the full version of [38].

Theorem 5.6.1. *For all $\delta > 0$, the following hold:*

- *Any value query interpolation mechanism that makes at most $\frac{e^{m^{1/3}}}{10m^8} - 1$ queries that obtains an approximation ratio $m^{1/3-\delta}/20$ whenever buyers have submodular valuations has first-round communication at least m^δ per bidder.*
- *Assuming $RP \neq NP$, any computationally efficient interpolation mechanism that obtains an approximation ratio $m^{1/3-\delta}/20$ has first-round communication at least m^δ per bidder.*

Corollary 5.6.1. *Assuming $RP \neq NP$, no a priori learnable computationally efficient mechanism or a priori learnable value query mechanism that makes at most $\frac{e^{m^{1/3}}}{10m^8} - 1$ queries guarantees a price of anarchy or price of stability $o(m^{1/3})$ at any solution concept that is guaranteed to exist on all profiles of submodular buyers.*

Part III

Rank Aggregation with Noisy Comparisons

Chapter 6

Top-k Ranking with Noisy Comparisons in Rounds

The results of this chapter are based on joint work with Mark Braverman and Matt Weinberg [34].

6.1 Introduction

Rank aggregation is a fundamental problem with numerous important applications, ranging from well-studied settings such as social choice [50] and web search [93] to newer platforms such as crowdsourcing [58] and peer grading [167]. Salient common features among these applications is that in the end, *ordinal* rather than *cardinal* information about the elements is relevant, and a precise fine-grained ordering of the elements is often unnecessary. For example, the goal of social choice is to select the best alternative, regardless of how good it is. In a curved course, the goal of peer grading is to partition assignments into quantiles corresponding to A/B/C/D, etc, regardless of their absolute quality.

Prior work has produced numerous ordinal aggregation procedures (i.e. based on comparisons of elements rather than cardinal evaluations of individual elements) in different settings, and we overview those most relevant to our work in Section 6.1.1. However, existing models from this literature fail to capture an important aspect of the problem with respect to some of the newer applications; that *multiple rounds of interaction are costly*. In crowdsourcing, for instance, one round of interaction is the time it takes to send out a bunch of tasks to users and wait for their responses before deciding which tasks to send out next, which is the main computational bottleneck. In peer grading, each round of interaction might take a week, and grades are expected to be determined certainly within a few weeks. In conference decisions, even one round of interaction seems to be pushing the time constraints.

Fortunately, the TCS community already provides a vast literature of algorithms with this constraint in mind, under the name of parallel algorithms. For instance, previous work resolves questions like "how many interactive rounds are necessary for a deterministic or randomized algorithm to select the k^{th} element with $O(n)$ total comparisons?" [186, 170, 3, 5, 6, 30]. This line of research, however, misses a different important aspect related to these applications (that is, in fact, captured by most works in rank aggregation), that the comparisons might be erroneous. Motivated by applications such as crowdsourcing and peer grading, we therefore study the round complexity of PARTITION, the problem of partitioning a totally ordered set into the top k and bottom $n-k$ elements, when comparisons might be erroneous.

Our first results on this front provide matching upper and lower bounds on what is achievable for PARTITION in just one round in three different models of error: noiseless (where the comparisons are correct), erasure (where comparisons are erased with probability $1-\gamma$), and noisy (where comparisons are correct with probability $1/2+\gamma/2$ and incorrect otherwise). We provide one-round algorithms using dn comparisons that make $O(n/d), O(n/(d\gamma))$, and $O(n/(d\gamma^2))$ mistakes (a mistake is any element placed on the wrong side of the partition) with high probability in the three models, respectively. The algorithms are randomized and different for each model, and the bounds hold both when d is an absolute constant or a function of n and γ. We provide asymptotically matching lower bounds as well: all (potentially randomized) one-round algorithms using dn comparisons necessarily make $\Omega(n/d), \Omega(n/(d\gamma))$, and

$\Omega(n/(d\gamma^2))$ mistakes in expectation in the three models, respectively. We further show that the same algorithms and lower bound constructions are also optimal (up to absolute constant factors) if mistakes are instead weighted by various different measures of their distance to k, the cutoff.[1]

After understanding completely the tradeoff between the number of comparisons and mistakes for one-round algorithms in each of the three models, we turn our attention to multi-round algorithms. Here, the results are more complex and can't be summarized in a few sentences. We briefly overview our multi-round results in each of the three models below. Again, *all* of the upper and lower bounds discussed below extend when mistakes are weighted by their distance to the cutoff. We overview the techniques used in proving our results in Section 6.1.2, but just briefly note here that the level of technicality roughly increases as we go from the noiseless to erasure to noisy models. In particular, lower bounds in the noisy model are quite involved.

Multi-Round Results in the Noiseless Model.

1. We design a 2-round algorithm for PARTITION using n/ε total comparisons that makes $O(n^{1/2+\varepsilon}\text{poly}(\log n)))$ mistakes with probability $1-e^{-\Omega(n)}$, and prove a nearly matching lower bound of $\Omega(\sqrt{n}\cdot\varepsilon^{5/2})$ mistakes, for any $\varepsilon > 0$ (ε may be a constant or a function of n).

2. We design a 3-round algorithm for PARTITION making $O(n\cdot\text{poly}(\log n))$ total comparisons that makes *zero* mistakes with probability $1-e^{-\Omega(n)}$. It is known that $\omega(n)$ total comparisons are necessary for a 3-round algorithm just to solve SELECT, the problem of *finding* the k^{th} element, with probability $1-o(1)$ [30].

3. We design a 4-round algorithm for PARTITION making $O(n)$ total comparisons that makes *zero* mistakes with probability $1-e^{-\Omega(n)}$. This matches the guarantee provided

[1]Specifically, if WRONG_i denotes the random variable that is 1 if an algorithm misplaces i and 0 otherwise, we consider measures of the following form, for any choice of c: $\sum_i \text{WRONG}_i|i-k|^c$. For example, $c=0$ counts the number of mistakes. This is further discussed in Section 6.2.

by an algorithm of Bollobás and Brightwell for SELECT, but is significantly simpler (in particular, it avoids any graph theory) [30].

Multi-Round Results in the Erasure Model.

1. We design a $O(\log^*(n))$-round algorithm for PARTITION making $O(n/\gamma)$ total comparisons that makes zero mistakes with probability $1 - e^{-\Omega(n)}$.

2. We show that no $o(\log^*(n))$-round algorithm even for SELECT making $O(n/\gamma)$ total comparisons can succeed with probability 2/3.

Multi-Round Results in the Noisy Model.

1. We design a 4-round algorithm for PARTITION making $O(n \log n/\gamma^2)$ comparisons that makes zero mistakes with high probability (a trivial corollary of our noiseless algorithm).

2. We show that no algorithm even for SELECT making $o(n \log n/\gamma^2)$ comparisons can succeed with probability 2/3 (in any number of rounds).

3. We design an algorithm for FINDMIN (the special case of SELECT with $k = n$) making $O(n/\gamma^2)$ comparisons that succeeds with probability 2/3. We also show that no algorithm making $o(n \log n/\gamma^2)$ comparisons can solve FINDMIN with probability $1 - 1/\text{poly}(n)$ (in any number of rounds).

Together, these results tell an interesting story. In one round, one can obtain the same guarantee in the noiseless versus erasure model with an additional factor of $1/\gamma$ comparisons. And one can obtain the same guarantee in the erasure versus noisy model with an additional factor of $1/\gamma$ comparisons. In some sense, this should be expected, because this exactly captures the degradation in information provided by a single comparison in each of the three models (a noiseless comparison provides one bit of information, an erasure comparison provides γ bits of information, and a noisy comparison provides $\Theta(\gamma^2)$ bits of information).

But in multiple rounds, everything changes. In four rounds, one can perfectly partition with high probability and $O(n)$ total comparisons in the noiseless model. In the erasure model, one can indeed partition perfectly with high probability and $O(n/\gamma)$ comparisons, but now it requires $\Theta(\log^*(n))$ rounds instead of just 4. Moreover, in the noisy model, any algorithm even solving SELECT with probability 2/3 requires an $\Omega(\log n/\gamma)$ blow-up in the number of comparisons, in any number of rounds! Note that neither of these additional factors come from the desire to succeed with high probability (as the lower bounds hold against even a 2/3 success) *nor* the desire to partition every element correctly (as the lower bounds hold even for just SELECT), but just from the way in which interaction helps in the three different models.

While we believe that the story told by our work as a whole provides the "main result," it is also worth emphasizing independently our results in the noisy model. Our one-round algorithm, for instance, is more involved than its counterparts in the noiseless and erasure models and our analysis uses the theory of biased random walks. Our multi-round lower bounds against SELECT and findMin in the noisy model are the most technical results of the paper, and tell their own interesting story about the difference between findMin and SELECT in the noisy model. To our knowledge, most tight lower bounds known for SELECT come directly from lower bounding findMin. It's surprising that findMin requires $\Theta(\log n)$ fewer comparisons than SELECT to solve with probability 2/3 in the noisy model.

We proceed now by discussing some related works below, and briefly overviewing our techniques in Section 6.1.2. We provide some conclusions and future directions in Section 6.1.3. Our single-round results are discussed in Section 6.3 and our multi-round results are discussed in Section 6.4.

6.1.1 Related Work

Rank aggregation is an enormous field that we can't possibly summarize in its entirety here. Some of the works most related to ours also study PARTITION (sometimes called TOP-K).

Almost all of these works also consider the possibility of erroneous comparisons, although sometimes under different models where the likelihood of an erroneous comparison scales with the distance between the two compared elements [61, 46, 95]. More importantly, to our knowledge this line of work either considers settings where the comparisons are exogenous (the designer has no control over which comparisons are queried, she can just analyze the results), or only analyze the query complexity and not the round complexity of designed algorithms. Our results contribute to this line of work by providing algorithms designed for settings like crowdsourcing or peer grading where the designer does have design freedom, but may be constrained by the number of interactive rounds.

There is a vast literature from the parallel algorithms community studying various sorting and selection problems in the noiseless model. For instance, tight bounds are known on the round complexity of Select for deterministic algorithms using $O(n)$ total comparisons (it is $\Theta(\log\log n)$) [186, 3], and randomized algorithms using $O(n)$ total comparisons (it is 4) [6, 5, 170, 30]. Similar results are known for sorting and approximate sorting as well [64, 7, 4, 113, 32, 31, 141]. Many of the designed deterministic algorithms provide *sorting networks*. A sorting network on n elements is a circuit whose gates are binary comparators. The depth of a sorting network is the number of required rounds, and the number of gates is the total number of comparisons. Randomized algorithms are known to require fewer rounds than deterministic ones with the same number of total comparisons for both sorting and selecting [5, 30].

In the noisy model, one can of course take any noiseless algorithm and repeat every comparison $O(\log n/\delta^2)$ times in parallel. To our knowledge, positive results that avoid this simple repetition are virtually non-existent. This is likely because a lower bound of Leighton and Ma [142] proves that in fact no sorting network can provide an asymptotic improvement (for complete sorting), and our lower bound (Theorem 6.4.7) shows that no randomized algorithm can provide an asymptotic improvement for Select. To our knowledge, no prior work studies parallel sorting algorithms in the erasure model. On this front, our work contributes by addressing some open problems in the parallel algorithms literature, but more importantly by providing the first parallel algorithms and lower bounds for Select in the erasure and noisy models.

There is also an active study of sorting in the noisy model [40, 41, 146] within the TCS community without concern for parallelization, but with concern for *resampling*. An algorithm is said to resample if it makes the same comparison multiple times. Clearly, an algorithm that doesn't resample can't possibly find the median exactly in the noisy model (what if the comparison between $n/2$ and $n/2+1$ is corrupted?). The focus of these works is designing poly-time algorithms to find the maximum-likelihood ordering from a set of $\binom{n}{2}$ noisy comparisons. Our work is fundamentally different from these, as we have asymptotically fewer than $\binom{n}{2}$ comparisons to work with, and at no point do we try to find a maximum-likelihood ordering (because we only want to solve Partition).

6.1.2 Tools and Techniques

Single Round Algorithms and Lower Bounds. Our single round results are guided by the following surprisingly useful observation: in order for an algorithm to possibly know that i exceeds the k^{th} highest element, i must at least be compared to some element between itself and k (as otherwise, the comparison results would be identical if we replaced i with an element just below k). Unsurprisingly, it is difficult to guarantee that many elements within n/d of k are compared to elements between themselves and k using only dn total comparisons in a single round, and this forms the basis for our lower bounds. Our upper bounds make use of this observation as well, and basically are able to guarantee that an element is correctly placed with high probability whenever it is compared to an element between itself and k. It's interesting that the same intuition is key to both the upper and lower bounds. We provide a description of the algorithms and proofs in Section 6.3.

In the erasure model, the same intuition extends, except that in order to have a non-erased comparison between i and an element between i and k, we need to make roughly

$1/\gamma$ such comparisons. This causes our lower bounds to improve by a factor of $1/\gamma$. In the noisy model, the same intuition again extends, although this time the right language is that we need to learn $\Omega(1)$ bits of information from comparisons of i to elements between i and k, which requires $\Omega(1/\gamma^2)$ such comparisons, and causes the improved factor of $1/\gamma^2$ in our lower bounds. Our algorithms in these two models are similar to the noiseless algorithm, but the analysis becomes necessarily more involved. For instance, our analysis in the noisy model appeals to facts about biased random walks on the line.

Multi-Round Algorithms and Lower Bounds. Our constant-round algorithms in the noiseless model are based on the following intuition: once we reach the point that we are only uncertain about $o(n)$ elements, we are basically looking at a fresh instance of PARTITION on a significantly smaller input size, except we're still allowed $\Theta(n)$ comparisons per round. Once we're only uncertain about only $O(\sqrt{n})$ elements, one additional round suffices to finish up (by comparing each element to every other one). The challenge in obtaining a four-round algorithm (as opposed to just an $O(1)$-round algorithm) is ensuring that we make significant enough gains in the first three rounds.

Interestingly, these ideas for constant-round algorithms in the noiseless model don't prove useful in the erasure or noisy models. Essentially the issue is that even after a constant number of rounds, we are unlikely to be confident that many elements are above or below k, so we can't simply recurse on a smaller instance. Still, it is quite difficult to discover a formal barrier, so our multi-round lower bounds for the erasure and noisy models are quite involved. We refer the reader to Section 6.4 for further details.

6.1.3 Conclusions

We study the problems of PARTITION and SELECT in settings where interaction is costly in the noiseless, erasure, and noisy comparison models. We provide matching (up to absolute constant factors) upper and lower bounds for one round algorithms in all three models, which also show that the number of comparisons required for the same guarantee degrade proportional to the information provided by a single comparison. We also provide matching upper and lower bounds for multi-round algorithms in all three models, which also show that the round and query complexity required for the same guarantee in these settings degrades worse than just by the loss in information when moving between the three comparison models. Finally, we show a separation between findMin and Select in the noisy model.

We believe our work motivates two important directions for future work. First, our work considers some of the more important constraints imposed on rank aggregation algorithms in applications like crowdsourcing or peer grading, but not all. For instance, some settings might require that every submission receives the same amount of attention (i.e. is a member of the same number of comparisons), or might motivate a different model of error (perhaps where mistakes aren't independent or identical across comparisons). It would be interesting to design algorithms and prove lower bounds under additional restrictions motivated by applications.

Finally, it is important to consider incentives in these applications. In peer grading, for instance, the students themselves are the ones providing the comparisons. An improperly designed algorithm might provide "mechanism design-type" incentives for the students to actively misreport if they think it will boost their own grade. Additionally, there are also "scoring rule-type" incentives that come into play: grading assignments takes effort! Without proper incentives, students may choose to put zero or little effort into their grading and just provide random information. We believe that using ordinal instead of cardinal information will be especially helpful on this front, as it is much easier to design mechanisms when players just make binary decisions, and it's much easier to understand how the noisy information provided by students scale with effort (in our models, it is simply that γ will increase with effort). It is therefore important to design mechanisms for applications like peer grading by building off of our algorithms.

6.2 Preliminaries and Notation

In this work, we study two problems, Select and Partition. Both problems take as input a randomly sorted, totally ordered set and an integer k. For simplicity of notation, we denote the i^{th} smallest element of the set as i. So if the input set is of size n, the input is exactly $[n]$. In Select, the goal is to output the (location of the) element k. In Partition, the goal is to partition the elements into the top k, which we'll call A for Accept and the bottom $n-k$, which we'll call R for Reject. Also for ease of notation, we'll state all of our results for $k = n/2$, the median.

We say an algorithm solves Select if it outputs the median, and solves Partition if it places correctly all elements above and below the median. For Select, we will say that an algorithm is a t-approximation with probability p if it outputs an element in $[n/2-t, n/2+t]$ with probability at least p. For Partition, we will consider a class of success measures, parameterized by a constant c, and say the c-weighted error associated with a specific partitioning into $A \sqcup R$ is equal to $\sum_{i>n/2} I(i \in R)i^c + \sum_{i<n/2} I(i \in A)i^c$.[2] Interestingly, in all cases we study, the same algorithm is asymptotically optimal for all c.

Query and Round Complexity. Our algorithms will be comparison-based. We study both the number of queries, and the number of adaptive rounds necessary to achieve a certain guarantee.[3] We may not always emphasize the runtime of our algorithms, but they all run in time poly(n).

Notation. We always consider settings where the input elements are a priori indistinguishable, or alternatively, that our algorithms randomly permute the input before making comparisons. When we write $x < y$, we mean literally that $x < y$ in the ground truth. In

[2]For instance, $c = 0$ counts the number of mistakes. $c = 1$ counts the number of mistakes, weighted by the distance of the mistaken element from the median. $c = 2$ is similar to mean-squared-error, etc.

[3]For example, an algorithm that makes Q queries one at a time, waiting for the result of previous queries before deciding which queries to make next has round complexity Q. An algorithm that makes all queries up front, without knowing any results has round complexity 1. We call protocols with round complexity 1 *non-adaptive*.

the noisy model, the results of comparisons may disagree with the underlying ordering, so we say that x beats y if a noisy comparison of x and y returned x as larger than y (regardless of whether or not $x > y$).

Models of Noise. We consider three comparison models, which return the following when $a > b$.

- Noiseless: Returns a beats b.
- Erasure: Returns a beats b with probability γ, and $\perp$ with probability $1 - \gamma$.
- Noisy: Returns a beats b with probability $1/2 + \gamma/2$, and b beats a with probability $1/2 - \gamma/2$.

Partition versus Select. We design all of our algorithms for PARTITION, and prove all of our lower bounds against SELECT. We do this because SELECT is in some sense a strictly easier problem than PARTITION. We discuss how one can get algorithms for SELECT via algorithms for PARTITION and vice versa formally in the full version of [34].

Resampling. Finally, note that in the erasure and noisy models, it may be desireable to query the same comparison multiple times. This is called *resampling*. It is easy to see that without resampling, it is impossible to guarantee that the exact median is found with high probability, even when all $\binom{n}{2}$ comparisons are made (what if the comparison between $n/2$ and $n/2 + 1$ is corrupted?). Resampling is not necessarily undesireable in the applications that motivate this work, so we consider our main results to be in the model where resampling is allowed. Still, it turns out that all of our algorithms can be easily modified to avoid resampling at the (necessary) cost of a small additional error, and it is easy to see the required modifications.[4] All of our lower bounds hold even against algorithms that resample.

[4]Essentially, replace all resampled comparisons with comparisons to "nearby" elements instead.

6.3 Results for Non-Adaptive Algorithms

In this section, we provide our results on non-adaptive (round complexity = 1) algorithms. We begin with the upper bounds below, followed by our matching (up to constant factors) lower bounds. Pesuedocodes and proofs of this section can be found in the full version of [34].

6.3.1 Upper Bounds

We provide asymptotically optimal algorithms in each of the three comparison models. Our three algorithms actually choose the same comparisons to make, but determine whether or not to accept or reject an element based on the resulting comparisons differently. The algorithms pick a *skeleton set* S of size $\sqrt{n}$ and compare every element in S to every other element. Each element not in S is compared to $d-1$ random elements of S.

From here, the remaining task in all three models is similar: the algorithm must first estimate the rank of each element in the skeleton set. Then, for each i, it must use this information combined with the results of $d-1$ comparisons to guess whether i should be accepted or rejected. The correct approach differs in the three models, which we discuss next.

Noiseless Model. First, we estimate that the median of the skeleton set, x, is close to the actual median. Then, we hope that each $i \notin S$ is compared to some element in S between itself and x. If this happens, we can pretty confidently accept or reject i. If it doesn't, then all we learn is that i is beaten by some elements above x and it beats some elements below x, which provides no helpful information about whether i is above or below the median, so we just make a random decision.

Theorem 6.3.1. *We have an algorithm which has query complexity dn, round complexity* 1*, does not resample, and outputs a partition that, for all c, has:*

- *expected c-weighted error* $O((n/d)^{c+1})$*, for any* $d = o(n^{1/4})$

- *c-weighted error* $O((n/d)^{c+1})$ *with probability* $1 - e^{-\Omega(n^3/d^{2c+2})}$*, for any* $d = o(n^{1/4})$*.*

The main ideas are the following. There are two sources of potential error in the algorithm. First, maybe the skeleton set is poorly chosen and not representative of the ground set. But this is extremely unlikely with such a large skeleton set. Second, note that if i is compared to any element in S between itself and x, *and* x is very close to $n/2$, then i will be correctly placed. If $|i - n/2| > n/d$, then we're unlikely to miss this window on $d - 1$ independent tries, and i will be correctly placed.

Erasure Model. At a high level, the algorithm for the erasure model is similar to the algorithm for the noiseless model.

Theorem 6.3.2. *We have an algorithm which has query complexity dn, round complexity* 1*, does not resample, and outputs a partition that, for all c, has:*

- *expected c-weighted error* $O((n/(d\gamma))^{c+1})$*, for any* d, γ *such that* $d/\gamma = o(n^{1/4})$

- *c-weighted error* $O((n/(d\gamma))^{c+1})$ *with probability* $1 - e^{-\Omega(n^3/d^{2c+2})}$*, whenever* $d/\gamma = o(\sqrt{n})$ *and* $d\gamma = o(n^{1/4})$*.*

The additional ingredient beyond the noiseless case is a proof that with high probability, not too many of the comparisons within S are erased and therefore while we can't learn the median of S exactly, we can learn a set of almost $|S|/2$ elements that are certainly above the median, and almost $|S|/2$ elements that are certainly below. If $i \notin S$ beats an element that is certainly above the median of S, we can confidently accept it, just like in the noiseless case.

Noisy Model. The algorithm for the noisy model is necessarily more involved than the previous two. We can still recover a good ranking of the elements in the skeleton set using the Braverman-Mossel algorithm [40], so this isn't the issue. The big difference between the noisy model and the previous two is that no single comparison can guarantee that $i \notin S$ should be accepted or rejected. Instead, every time we have a set of elements all above the median of S, x, of which i beats at least half, this provides some evidence that i should be accepted. Every time we have a set of elements all below x of which i is beaten by at least half, this provides some evidence that i should be rejected. The trick is now just deciding which evidence is stronger.

Theorem 6.3.3. *We have an algorithm which has query complexity dn, round complexity 1, does not resample, and outputs a partition that, for all c, has:*

- *expected c-weighted error $O((n/(d\gamma^2))^{c+1})$, for any $d = o(n^{1/4})$, $\gamma = \omega(n^{1/8})$.*
- *c-weighted error $O((n/(d\gamma^2))^{c+1})$ with probability $1 - e^{\Omega(n^3/d^{2c+2})}$, for any $d = o(n^{1/4})$, $\gamma = \omega(n^{1/8})$.*

6.3.2 Lower Bounds

In this section, we show that the algorithms designed in the previous section are optimal up to constant factors. All of the algorithms in the previous section are "tight," in the sense that we expect element i to be correctly placed whenever it is compared to enough elements between itself and the median. In the noiseless model, one element is enough. In the erasure model, we instead need $\Omega(1/\gamma)$ (to make sure at least one isn't erased). In the noisy model, we need $\Omega(1/\gamma^2)$ (to make sure we get $\Omega(1)$ bits of information about the difference between i and the median). If we don't have enough comparisons between i and elements between itself and the median, we shouldn't hope to be able to classify i correctly, as the comparisons involving i would look nearly identical if we replaced i with an element just on the other side of the median.

Theorem 6.3.4. *For all c, $d > 0$, any non-adaptive algorithm with query complexity dn necessarily has expected c-weighted error $\Omega((n/d)^{c+1})$ in the noiseless model, $\Omega((n/(d\gamma))^{c+1})$ in the erasure model, and $\Omega((n/(d\gamma^2))^{c+1})$ in the noisy model.*

6.4 Results for Multi-Round Algorithms

Pesuedocodes and proofs of this section can be found in the full version of [34].

6.4.1 Noiseless Model

We first present our algorithm and nearly matching lower bound for 2-round algorithms. The first round of our algorithm tries to get as good of an approximation to the median as possible, and then compares it to every element in round two. Getting the best possible approximation is actually a bit tricky. For instance, simply finding the median of a skeleton set of size $\sqrt{n}$ only guarantees an element within $\Theta(n^{3/4})$ of the median.[5] We instead take several "iterations" of nested skeleton sets to get a better and better approximation to the median. In reality, all iterations happen simultaneously in the first round, but it is helpful to think of them as sequential refinements.

For any $r \geq 1$, our algorithm starts with a huge skeleton set S_1 of $n^{2r/(2r+1)}$ random samples from $[n]$. This is too large to compare every element in S_1 with itself, so we choose a set $T_1 \subseteq S_1$ of $n^{1/(2r+1)}$ random pivots. Then we compare every element in S_1 to every element in T_1, and we will certainly learn two pivots, a_1 and b_1 such that the median of S_1 lies in $[a_1, b_1]$, and a p_1 such that the median of S_1 is exactly the $(p_1|A_1|)^{th}$ element of $A_1 = S_1 \cap [a_1, b_1]$. Now, we recurse within A_1 and try to find the $(p_1|A_1|)^{th}$ element. Of course, because all of these comparisons happen in one round, we don't know ahead of time in which subinterval of S_1 we'll want to recurse, so we have to waste a bunch of comparisons. These continual refinements still make some progress, and allow us to find a smaller and smaller window containing the median of S_1, which is a very good approximation to the true median because S_1 was so large.

[5]This is exactly what Bollobás and Brightwell do in the first round of their 4-round algorithm, which is why sophisticated graph theory follows to fit into four rounds. Our improved first round simplifies the remaining rounds.

Theorem 6.4.1. *For all c, r and $\varepsilon > 0$, we have an algorithm which has round complexity 2, query complexity $(r+1)n$, and outputs a partition that:*

- *has expected c-weighted error at most $(8rn^{(r+1)/(2r+1)+\varepsilon})^{c+1}$*
- *has c-weighted error at most $(8rn^{(r+1)/(2r+1)+\varepsilon})^{c+1}$ with probability at least $1 - re^{-n^{\Omega(\varepsilon)}}$.*

Note that setting $r = \log n$, and ε such that $n^{\varepsilon} = 8\log^3 n$, we get an algorithm with round complexity 2, query complexity $n\log n + n$ that outputs a partition with c-weighted error $O((\sqrt{n}\log^4 n)^{c+1})$ with probability $1 - O(\log n/n^2)$.

We also prove a nearly matching lower bound on two-round algorithms in the noiseless model. At a *very* high level, our lower bound repeats the argument of our one round lower bound twice. Specifically, we show that after one round, there are many elements within a window of size $\Theta(n/d)$ of the median such that a constant fraction of these elements have not been compared to any other elements in this window. We then show that after the second round, conditioned on this, there is necessarily a window of size $\approx \sqrt{n}$ such that a constant fraction of these elements have not been compared to any other elements in this window. Finally we show that this implies that we must err on a constant fraction of these elements. The actual proof is technical, but follows this high level outline.

Theorem 6.4.2. *For all c, and any $d = o(n^{1/5})$, any algorithm with query complexity dn and round complexity 2 necessarily has expected c-weighted error $\Omega((\sqrt{n}/d^{5/2})^{c+1})$.*

From here we show how to make use of our two-round algorithm to design a three-round algorithm that makes *zero* mistakes with high probability. After our two-round algorithm with appropriate parameters, we can be pretty sure that the median lies somewhere in a range of $O(\sqrt{n}\log^4 n)$, so we can just compare all of these elements to each other in one additional round.

Theorem 6.4.3. *For all c, we have an algorithm which has query complexity $O(n\log^8 n)$, round complexity 3, and outputs a partition with* zero *c-weighted error with probability $1 - O(\log n/n^2)$.*

Again, recall that $\omega(n)$ queries are necessary for any three-round algorithm just to solve SELECT with probability $1 - o(1)$ [30]. Finally, we further make use of ideas from our two-round algorithm to design a simple four round algorithm that has query complexity $O(n)$ and makes *zero* mistakes with high probability. More specifically, we appropriately tune the parameters for our two-round algorithm (i.e. set $r = 1$) to find a window of size $\approx n^{2/3}$ that contains the median (and already correctly partition all other elements). We then use similar ideas in round three to further find a window of size $\approx \sqrt{n}$ that contains the median (and again correctly partition all other elements). We use the final round to compare all remaining uncertain elements to each other and correctly partition them.

Theorem 6.4.4. *For all c, and any $\varepsilon \in (0, 1/18)$, we have an algorithm which has query complexity $O(n)$, round complexity 4, and outputs a partition with* zero *c-weighted error with probability at least $1 - e^{-\Omega(n^\varepsilon)}$.*

6.4.2 Erasure and Noisy Models

Here we briefly overview our results on multi-round algorithms in the erasure and noisy models. We begin with an easy reduction from these models to the noiseless model, at the cost of a blow-up in the round or query complexity. Essentially, we are just observing that one can adaptively resample any comparison in the erasure model until it isn't erased (which will take $1/\gamma$ resamples in expectation), and also that one can resample in parallel any comparison in either the erasure or noisy model the appropriate number of times and have it effectively be a noiseless comparison.

Proposition 6.4.1. *If there is an algorithm solving* PARTITION*,* SELECT *or* FINDMIN *in the noiseless model with probability p that has query complexity Q and round complexity r, then there are also algorithms that resample that:*

- *solve* Partition*,* Select *or* findMin *in the erasure model with probability* p *that have expected query complexity* Q/γ*, but perhaps with expected round complexity* Q/γ *as well.*

- *solve* Partition*,* Select *or* findMin *in the erasure model with probability* $p - 1/\mathrm{poly}(n)$ *that have query complexity* $O(Q(\log Q + \log n)/\gamma)$*, and round complexity* r*.*

- *solve* Partition*,* Select *or* findMin *in the noisy model with probability* $p - 1/\mathrm{poly}(n)$ *that have query complexity* $O(Q(\log Q + \log n)/\gamma^2)$*, and round complexity* r*.*

Corollary 6.4.1. *There are algorithms that resample that:*

- *solve* Partition *or* Select *in the erasure model with probability* 1 *with expected query complexity* $O(n/\gamma)$ *(based on the QuickSelect or Median-of-Medians algorithm [121, 29]).*

- *solve* Partition *or* Select *in the erasure model with probability* $1 - 1/\mathrm{poly}(n)$ *with query complexity* $O(n \log n/\gamma)$ *and round complexity* 4*.*

- *solve* Partition *or* Select *in the noisy model with probability* $1 - 1/\mathrm{poly}(n)$ *with query complexity* $O(n \log n/\gamma^2)$ *and round complexity* 4*.*

In the erasure model, the algorithms provided by this reduction do not have the optimal round/query complexity. We show that $\Theta(n/\gamma)$ queries are necessary and sufficient, as well as $\Theta(\log^*(n))$ rounds. For the algorithm, we begin by finding the median of a random set of size $n/\log n$ elements. This can be done in 4 rounds and $O(n/\delta)$ total comparisons by Corollary 6.4.1. Doing this twice in parallel, we find two elements that are guaranteed to be above/below the median, but very close. Then, we spend $\log^*(n)$ rounds comparing every element to both of these. It's not obvious that this can be done in $\log^*(n)$ rounds. Essentially what happens is that after each round, a fraction of elements are successfully compared, and we don't need to use any future comparisons on them. This lets us do even more comparisons involving the remaining elements in future rounds, so the fraction of successes actually increases with successive rounds. Analysis shows that the number of required rounds is therefore $\log^*(n)$ (instead of $\log(n)$ if the fraction was constant throughout all rounds). After this, we learn for sure that the median lies within a sublinear window, and we can again invoke the 4-round algorithm of Corollary 6.4.1 to finish up. Our lower bound essentially shows that it takes $\log^*(n)$ rounds just to have a non-erased comparison involving all n elements even with $O(n/\delta)$ per round, and that this implies a lower bound.

Theorem 6.4.5. *We have an algorithm which has query complexity $O(n/\gamma)$, round complexity $\log^*(n) + 8$, and solves* PARTITION *with probability at least $1 - 1/\text{poly}(n)$.*

Theorem 6.4.6. *Assume $\gamma \leq 1/2$. In the erasure model, any algorithm solving* SELECT *with probability 2/3 even with $O(n/\gamma)$ comparisons per round necessarily has round complexity $\Omega(\log^*(n))$.*

We now introduce a related problem that is strictly easier than PARTITION or SELECT, which we call RANK, and prove lower bounds on the round/query complexity of RANK noisy models, which will imply lower bounds on PARTITION and SELECT. In RANK, we are given as input a set S of n elements, and a special element b and asked to determine b's rank in S (i.e. how many elements in S are less than b). We say that a solution is a t-approximation if the guess is within t of the element's actual rank. We show formally that RANK is strictly easier than SELECT in the full version of [34]. From here, we prove lower bounds against RANK in the noisy model.

At a high level, we show (in the proof of Theorem 6.4.7) that with only $O(n \log n/\gamma^2)$ queries, it's very likely that there are a constant fraction of a_i's such that the algorithm is can't be very sure about the relation between a_i and b. This might happen, for instance, if not many comparisons were done between a_i and b and they were split close to 50-50. From here, we use an anti-concentration inequality (the Berry-Essen inequality) to show that the rank of b does not concentrate within some range of size $\Theta(n^{3/8})$ conditioned on the available information. In otherwords, the information available simply cannot narrow down the rank of b to within a small window with decent probability, no matter how that information is used. We then conclude that no algorithms with $o(n \log n/\gamma^2)$ comparisons can approximate the rank well with probability 2/3.

Theorem 6.4.7. *In the noisy model, any algorithm obtaining an* $(n^{3/8}/40)$*-approximation for* Rank *with probability* $2/3$ *necessarily has query complexity* $\Omega(n \log n/\gamma^2)$*.*

Finally, we conclude with an algorithm for findMin in the noisy model showing that findMin is strictly easier than Select. This is surprising, as most existing lower bounds against Select are obtained by bounding findMin. Our algorithm again begins by finding the minimum, x, of a random set of size $n/\log n$ using $O(n/\gamma^2)$ total comparisons by Corollary 6.4.1. Then, we iteratively compare each element to x a fixed number of times, throwing out elements that beat it too many times. Again, as we throw out elements, we get to compare the remaining elements to x more and more. We're able to show that after only an appropriate number of iterations (so that only $O(n/\delta^2)$ total comparisons have been made), it's very likely that only $n/\log n$ elements remain, and that with constant probability the true minimum was not eliminated. From here, we can again invoke the algorithm of Corollary 6.4.1 to find the true minimum (assuming it wasn't eliminated).

Theorem 6.4.8. *Assume n is large enough and* $10 \leq c \leq \log n$*. We have an algorithm which has query complexity* $\frac{3cn}{\gamma^2}$ *and solves* findMin *in the noisy model with probability at least* $1 - e^{-\Omega(c)}$*.*

Theorem 6.4.9. *Assume* $c \geq 1$*, n is large enough and* $\gamma \leq 1/4$*. Any algorithm in the noisy model with query complexity* $\frac{cn}{\gamma^2}$ *solves* findMin *with probability at most* $1 - e^{-O(c)}$*.*

Theorem 6.4.8 shows that findMin is strictly easier than Select (as it can be solved with constant probability with asymptotically fewer comparisons). Theorem 6.4.9 is included for completeness, and shows that it is not possible to get a better success probability without a blow-up in the query complexity.

Chapter 7

Top-k Ranking under the Strong Stochastic Transitivity Model

The results of this chapter are based on joint work with Xi Chen, Sivakanth Gopi and Jon Schneider [57].

7.1 Introduction

The problem of inferring a ranking over a set of n items, such as documents, images, movies, or URL links, is an important problem in machine learning and finds many applications in recommender systems, web search, social choice, and many other areas. One of the most popular forms of data for ranking is pairwise comparison data, which can be easily collected via, for example, crowdsourcing, online games, or tournament play. The problem of ranking aggregation from pairwise comparisons has been widely studied and most work aims at inferring a total ordering of all the items (see, e.g., [159]). However, for some applications with a large number of items (e.g., rating of restaurants in a city), it is only necessary to identify the set of top K items. For these applications, inferring the total global ranking order unnecessarily increases the complexity of the problem and requires significantly more samples. Typically, the sample complexity of recovering the set of top K items is inversely related to the gap between item K and item $K+1$. On the other hand, the sample complexity of recovering the global ranking order might depend on the the minimum of the gaps between two consecutive items.

In the basic setting for this problem, there is a set of n items with some true underlying ranking. For possible pair (i, j) of items, an analyst is given r noisy pairwise comparisons between those two items, each independently ranking i above j with some probability p_{ij}. From this data, the analyst wishes to identify the top K items in the ranking, ideally using as few samples r as is necessary to be correct with sufficiently high probability. The noise in the pairwise comparisons (i.e., the probabilities p_{ij}) is constrained by the choice of noise model. Many existing models - such as the Bradley-Terry-Luce model (BTL) [33, 145], the Thurstone model [183], and their variants - are *parametric* comparison models, in that each probability p_{ij} is of the form $f(s_i, s_j)$, where s_i is a 'score' associated with item i. While these parametric models yield many interesting algorithms with provable guarantees [60, 125, 179], the models enforce strong assumptions on the probabilities of incorrect pairwise comparisons that might not hold in practice [69, 150, 185, 24].

A more general class of pairwise comparison model is the strong stochastic transitivity (SST) model, which subsumes the aforementioned parameter models as special cases and has a wide range of applications in psychology and social science (see, e.g., [69, 150, 102]). The SST model only enforces the following coherence assumption: if i is ranked above j, then $p_{il} \geq p_{jl}$ for all other items l. [175] pioneered the algorithmic and theoretical study of ranking aggregation under SST models. For top-K ranking problems, [177] proposed a counting-based algorithm under a very general noise model that includes SST as a special case. The algorithm simply orders the items by the total number of pairwise comparisons won. For a certain class of instances, this algorithm is in fact optimal; any algorithm with a constant probability of success on these instances needs roughly at least as many samples as this counting algorithm. However, this does not rule out the existence of other instances where the counting algorithm performs asymptotically worse than some other algorithm (see the example in Eq. (7.1)).

Under the SST model, we study algorithms for the top-K problem from the standpoint of *instance-specific analysis* (a.k.a. *competitive analysis* in the computer science). This is in spirit very similar to the notion "instance optimal"[96]. We give an algorithm which, on any instance, needs at most $\tilde{O}(\sqrt{n})$ times as many samples as the best possible algorithm for that instance to succeed with the same probability. We further show this result is tight (up to polylogarithmic factors): for any algorithm, there are instances where that algorithm needs at least $\tilde{\Omega}(\sqrt{n})$ times as many samples as the best possible algorithm. In contrast, the counting algorithm of [177] sometimes requires $\Omega(n)$ times as many samples as the best possible algorithm, even when the probabilities p_{ij} are bounded away from 1.

Our main technical tool is the introduction of a new decision problem we call *domination*, which captures the difficulty of solving the top-K problem while being simpler to directly analyze via information theoretic techniques. The domination problem can be thought of as a restricted one-dimensional variant of the top-K problem, where the analyst is only given the outcomes of pairwise comparisons that involve item i or j, and wishes to determine whether i is ranked above j. Our proof of the above claims proceeds by proving analogous competitive ratio results for the domination problem, and then carefully embedding the domination problem as part of the top-K problem. To establish the competitive ratio for the domination, we start from a simple case where the comparison probabilities are bounded away from zero and one. We first show that a popular counting algorithm developed by [177] has a sub-optimal competitive ratio of $\tilde{\Theta}(n)$. The main reason is that the counting algorithm treats samples from different coordinates of comparison probability vector equally. To address the issue of the counting algorithm, another *maximum algorithm* is first proposed. However, the maximum algorithm still leads to a sub-optimal competitive ratio and it fails whenever the counting algorithm performs well. Therefore, we develop techniques to combine the counting and maximum algorithms together, which give the optimal competitive ratio of $\tilde{O}(\sqrt{n})$. More detailed description of this idea is provided in Section 7.3.1.

7.1.1 Related Work

The problem of sorting a set of items from a collection of pairwise comparisons is one of the most classical problems in computer science and statistics. Many works investigate the problem of recovering the total ordering under noisy comparisons drawn from some parametric model. For the BTL model, Negahban et al. [159] propose the *RankCentrality* algorithm, which serves as the building block for many spectral ranking algorithms. Lu and Boutilier [144] give an algorithm for sorting in the Mallows model. Rajkumar and Agarwal [169] investigate which statistical assumptions (BTL models, generalized low-noise condition, etc.) guarantee convergence of different algorithms to the true ranking.

More recently, the problem of top-K ranking has received a lot of attention. Chen and Suh [60], Jang et al. [125], and Suh et al. [179] all propose various spectral methods for the BTL model or a mixture of BTL models. Eriksson [95] considers a noisy observation model where comparisons deviating from the true ordering are *i.i.d.* with bounded probability. In [177], Shah and Wainwright propose a counting-based algorithm, which motivates our work. However, their algorithm is not instance adaptive and we provide a simple instance (see Eq. (7.1)) illustrating that the sample complexity in [177] is sub-optimal on that instance. The top-K ranking problem is also related to the best K arm identification in multi-armed bandit [44, 123, 191]. However, in the latter problem, the samples are *i.i.d.* random variables rather than pairwise comparisons and the goal is to identify the top K distributions with largest means.

This paper and the above references all belong to the *non-active* setting: the set of data provided to the algorithm is fixed, and there is no way for the algorithm to adaptively choose additional pairwise comparisons to query. In several applications, this property is desirable, specifically if one is using a well-established dataset or if adaptivity is costly (e.g., on some crowdsourcing platforms). Nonetheless, the problems of sorting and top-K ranking are incredibly interesting in the adaptive setting as well. Several works [1, 124, 130, 35] consider the adaptive noisy sorting problem with (noisy) pairwise comparisons and explore the sample complexity to recover an (approximately) correct total ordering in terms of some distance function (e.g,., Kendall's tau). In [189], Wauthier et al. propose simple weighted counting algorithms to recovery an approximate total ordering from noisy pairwise comparisons. Dwork et al. [92] and Ailon et al. [2] consider a related *Kemeny optimization* problem, where the goal is to determine the total ordering that minimizes the sum of the distances to different permutations. More recently, the top-K ranking problem in the active setting has been studied by Braverman et al. [34] where they consider the tradeoff between the sample complexity of algorithms and the number of rounds of adaptivity. All of this work takes place in much more constrained noise models than the SST model. A very recent work by Heckel et. al. [119] investigates the active ranking under a general class of nonparametric models and also establishes a lower bound on the number of comparisons for parametric models. However, developing an active instance-adaptive ranking algorithm under the SST model still remains an interesting open problem.

The instance adaptivity has been widely studied in many statistical estimation problems. For example, the adaptive estimation is an important topic in nonparametric shape-restricted regression (see, e.g., [112, 53, 54, 111]). Shah et. al. [176] study the adaptive estimation problem for estimating comparison probabilities in a SST model. The concept of instance adaptivity is also closely related to the oracle inequality, which relates the performance of a constructed estimator with that of an "oracle" estimator with the information about local structure of the parameter space (see the survey paper [49] and the book [132] and references therein).

[177] discussed the approximate recovery of top items. The approximate recovery would be suitable for many practical applications. In their paper, they showed that this approximate relaxation allows a less constrained separation threshold. For our algorithms, it is not

clear that the approximate relaxation can significantly improve the competitive ratios. It is an interesting open question to extend our work to see if the approximate recovery can result in better competitive ratios.

7.2 Preliminaries and Problem Setup

7.2.1 The Top-K problem

Consider the following problem. An analyst is given a collection of n items, labelled 1 through n. These items have some true ordering defined by a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that for $1 \leq u < v \leq n$, the item labelled $\pi(u)$ has a better rank than the item labelled $\pi(v)$ (i.e., the item with label i has a better rank than the item j if and only if $\pi^{-1}(i) < \pi^{-1}(j)$). The analyst's goal is to determine the set of the top K items, i.e., $\{\pi(1), \ldots, \pi(k)\}$.

The analyst receives r samples. Each sample consists of pairwise comparisons between all pairs of items. All the pairwise comparisons are independent with each other. The outcomes of the pairwise comparison between any two items is characterized by the probability matrix $\mathbf{P} \in [0,1]^{n\times n}$. For a pair of items (i, j), let $X_{i,j} \in \{0, 1\}$ be the outcome of the comparison between the item i and j, where $X_{i,j} = 1$ means i is preferred to j (denoted by $i \succ j$) and $X_{i,j} = 0$ otherwise. Further, let $\mathcal{B}(z)$ denote the Bernoulli random variable with mean $z \in [0, 1]$. The outcome $X_{i,j}$ follows $\mathcal{B}(\mathbf{P}_{\pi^{-1}(i),\pi^{-1}(j)})$, i.e.,

$$\Pr(X_{i,j} = 1) = \Pr(i \succ j) = \mathbf{P}_{\pi^{-1}(i),\pi^{-1}(j)}.$$

The probability matrix $\mathbf{P}$ is said to be strong stochastic transitive (SST) if it satisfies the following definition.

Definition 7.2.1. *The $n \times n$ probability matrix $\mathbf{P} \in [0,1]^{n\times n}$ is strong stochastic transitive (SST) if*

1. *For* $1 \leq u < v \leq n$, $\mathbf{P}_{u,l} \geq \mathbf{P}_{v,l}$ *for all* $l \in [n]$.

2. $\mathbf{P}$ *is shifted-skew-symmetric (i.e.,* $\mathbf{P} - 0.5$ *is skew-symmetric) where* $\mathbf{P}_{v,u} = 1 - \mathbf{P}_{u,v}$ *and* $\mathbf{P}_{u,u} = 0.5$ *for* $u \in [n]$.

The first condition claims that when the item i has a higher rank than item j (i.e., $\pi^{-1}(i) < \pi^{-1}(j)$), for any other item k, we have

$$\Pr(i \succ k) = \mathbf{P}_{\pi^{-1}(i),\pi^{-1}(k)} \geq \Pr(j \succ k) = \mathbf{P}_{\pi^{-1}(j),\pi^{-1}(k)}.$$

Remark 7.2.1. *Many classical parametric models such that BTL [33, 145] and Thurstone (Case V) [183] models are special cases of SST. More specifically, parametric models assume a score vector* $w_1 \geq w_2 \geq \ldots \geq w_n$. *They further assume that the comparison probability* $\mathbf{P}_{u,v} = F(w_u - w_v)$, *where* $F : \mathbb{R} \to [0, 1]$ *is a non-decreasing function and* $F(t) = 1 - F(-t)$ *(e.g.,* $F(t) = 1/(1+\exp(-t))$ *in BTL models). By the property of* F*, it is easy to verify that* $\mathbf{P}_{u,v} = F(w_u - w_v)$ *satisfy the conditions in Definition 7.2.1.*

Under the SST models, we can formally define the top-K ranking problem as follows. The top-K ranking problem takes the inputs n, k, r that are known to the algorithm and the SST probability matrix $\mathbf{P}$ that is unknown to the algorithm.

Definition 7.2.2. Top-$k(n, k, \mathbf{P}, r)$ *is the following algorithmic problem:*

1. *A permutation* π *of* $[n]$ *is uniformly sampled.*

2. *The algorithm is given samples* $X_{i,j,l}$ *for* $i \in [n], j \in [n], l \in [r]$, *where each* $X_{i,j,l}$ *is sampled independently according to* $\mathcal{B}(\mathbf{P}_{\pi^{-1}(i),\pi^{-1}(j)})$. *The algorithm is also given the value of* k, *but not* π *or the matrix* $\mathbf{P}$.

3. *The algorithm succeeds if it correctly outputs the set of labels* $\{\pi(1), ..., \pi(k)\}$ *of the top* k *items.*

Remark 7.2.2. *We note that [177] considers a slightly different observation model in which each pair is queried r times. For each query, one can obtain a comparison result with the probability $p_{obs} \in (0,1]$ and with probability $1 - p_{obs}$, the query is invalid. In this model, each pair will be compared $r \cdot p_{obs}$ times on expectation. When $p_{obs} = 1$, it reduces to our model in Definition 7.2.2, where we observe exactly r comparisons for each pair. Our results can be easily extended to deal with the observation model in [177] by replacing r with the effective sample size, $r \cdot p_{obs}$. We omit the details for the sake of simplicity.*

Our primary metric of concern is the *sample complexity* of various algorithms; that is, the minimum number of samples an algorithm A requires to succeed with a given probability. To this end, we call the triple $S = (n, k, \mathbf{P})$ an *instance* of the TOP-k problem, and write $r_{min}(S, A, p)$ to denote the minimum value such that for all $r \geq r_{min}(S, A, p)$, A succeeds on instance S with probability p when given r samples. When p is omitted, we will take $p = \frac{3}{4}$; i.e., $r_{min}(S, A) = r_{min}(S, A, \frac{3}{4})$. It is worthwhile to note that, by repeating the algorithm constant number of times and taking the majority output, solving the problem for any constant error translates to a solution with polynomially decaying error, and the sample complexity will increase only by a multiplicative logarithmic factor.

7.2.2 The Domination problem

To solve the problem of TOP-k, we study a key sub-problem called DOMINATION, which captures the core of the difficulty of TOP-k. In particular, DOMINATION captures the dominance relation between two consecutive rows of a SST probability matrix. DOMINATION is formally defined as follows.

Definition 7.2.3. DOMINATION$(n, \mathbf{p}, \mathbf{q}, r)$ *is the following algorithmic problem:*

1. $\mathbf{p} = (p_1, \cdots, p_n)$ *and* $\mathbf{q} = (q_1, \cdots, q_n)$ *are two vectors of probabilities that satisfy* $1 \geq p_i \geq q_i \geq 0$ *for all* $i \in [n]$. $\mathbf{p}, \mathbf{q}$ *are not given to the algorithm.*

2. *A random bit B is sampled from $\mathcal{B}(\frac{1}{2})$. Samples $X_{i,j}, Y_{i,j}$ (for $i \in [n], j \in [r]$) are generated as follows:*

   (a) *Case $B = 0$: each $X_{i,j}$ is independently sampled according to $\mathcal{B}(p_i)$ and each $Y_{i,j}$ is independently sampled according to $\mathcal{B}(q_i)$.*

   (b) *Case $B = 1$: each $X_{i,j}$ is independently sampled according to $\mathcal{B}(q_i)$ and each $Y_{i,j}$ is independently sampled according to $\mathcal{B}(p_i)$.*

   The algorithm is given the samples $X_{i,j}$ and $Y_{i,j}$, but is not given the bit B or the values of $\mathbf{p}$ and $\mathbf{q}$.

3. *The algorithm succeeds if it correctly outputs the value of the hidden bit B.*

From Definition 7.2.1, it is clear for any pair of rows (or columns) of a SST probability matrix $\mathbf{P}$, one row (or column) will dominate another. As before, we are interested in the sample complexity of algorithms for Domination. We call the triple $C = (n, \mathbf{p}, \mathbf{q})$ an instance of Domination, and write $r_{min}(C, A, p)$ to be the minimum value such that for all $r \geq r_{min}(C, A, p)$, algorithm A succeeds at solving Domination$(n, \mathbf{p}, \mathbf{q}, r)$ with probability at least p. Moreover, for notational simplicity, let $r_{min}(C, A) = r_{min}(C, A, \frac{3}{4})$.

There are at least two main approaches one can take to analyze the sample complexity of problems like Top-k or Domination. The first (and more common) approach is to bound the value of $r_{min}(S, A)$ by some explicit function $f(S)$ of a Top-k instance S. This is the approach taken by [177]. They show that for some simple function f (roughly, the square of the reciprocal of the absolute difference of the sums of the k-th and $(k+1)$-th rows of the matrix $\mathbf{P}$ i.e. $1/\|\mathbf{P}_k - \mathbf{P}_{k+1}\|_1^2$), there is an algorithm A such that for all instances S, $r_{min}(S, A) = O(f(S))$; moreover this is optimal in the sense that there exists an instance S such that for all algorithms A, $r_{min}(S, A) = \Omega(f(S))$. While this is a natural approach, it leaves open the question of what the correct choice of f should be; indeed, different choices of f give rise to different 'optimal' algorithms A which outperform each other on different instances.

In this paper, we take the second approach, which is to compare the sample complexity of an algorithm on an instance to the sample complexity of the best possible algorithm on that instance. Formally, let $r_{min}(S,p) = \inf_A r_{min}(S,A,p)$ and let $r_{min}(S) = r_{min}(S, \frac{3}{4})$. An ideal algorithm A would satisfy $r_{min}(S,A) = \Theta(r_{min}(S))$ for all instances S of Top-k; more generally, we are interested in bounding the ratio between $r_{min}(S,A)$ and $r_{min}(S)$. We call this ratio the *competitive ratio* of the algorithm, and say that an algorithm is $f(n)$-competitive if $r_{min}(S,A) \le f(n) r_{min}(S)$. We likewise define the corresponding notions for Domination.

7.3 Main Results

In our main upper bound result, we give a linear-time algorithm for Top-k which is $\tilde{O}(\sqrt{n})$-competitive (restatement of Corollary 7.7.1):

Theorem 7.3.1. *There is an algorithm A for* Top-k *such that A runs in time $O(n^2 r)$ and on every instance S of* Top-k *on n items,*

$$r_{min}(S,A) \le O(\sqrt{n}\log n) r_{min}(S).$$

In our main lower bound result, we show that up to logarithmic factors, this $\sqrt{n}$ competitive ratio is optimal (restatement of Theorem 7.8.1):

Theorem 7.3.2. *For any algorithm A for* Top-k*, there exists an instance S of* Top-k *on n items such that*

$$r_{min}(S,A) \ge \Omega\left(\frac{\sqrt{n}}{\log n}\right) r_{min}(S).$$

In comparison, for the counting algorithm A' of [177], there exist instances S such that $r_{min}(S,A') \ge \tilde{\Omega}(n) r_{min}(S)$. For example, consider the instance $S = (n, k, \mathbf{P})$ with

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2}+\varepsilon & \cdots & \cdots & \frac{1}{2}+\varepsilon \\ \frac{1}{2}-\varepsilon & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \frac{1}{2}+\varepsilon \\ \frac{1}{2}-\varepsilon & \cdots & \cdots & \frac{1}{2}-\varepsilon & \frac{1}{2} \end{bmatrix} \tag{7.1}$$

It is straightforward to show that with $\Theta(\log n/\varepsilon^2)$ samples, we can learn all pairwise comparisons correctly with high probability by taking a majority vote, and therefore even sort all the elements correctly. This implies that $r_{min}(S) = O(\log n/\varepsilon^2)$. On the other hand, we show in Corollary 7.5.1 that $r_{min}(S, A') = \Omega(n/\varepsilon^2)$ when $\varepsilon < 1/10$.

7.3.1 Main Techniques and Overview

We prove our main results by first proving similar results for DOMINATION which we defined in Definition 7.2.3. Intuitively DOMINATION captures the main hardness of TOP-k while being much simpler to analyze. Once we prove upper bound and lower bounds for the sample complexity of DOMINATION, we will use reductions to prove analogous results for TOP-k.

We begin in Section 7.4, by proving a general lower bound on the sample complexity of domination. Explicitly, for a given instance $C = (n, \mathbf{p}, \mathbf{q})$ of DOMINATION, we show that $r_{min}(C) \geq \Omega(1/\mathbb{I}(\mathbf{p}, \mathbf{q}))$ where $\mathbb{I}(\mathbf{p}, \mathbf{q})$ is the amount of information we can learn about the bit B from one sample of pairwise comparison in each of the coordinates.

In Section 7.5, we proceed to design algorithms for DOMINATION restricted to instances $C = (n, \mathbf{p}, \mathbf{q})$ where $\delta \leq p_i, q_i \leq 1 - \delta$ for some constant $0 < \delta \leq 1/2$. In this regime $\mathbb{I}(\mathbf{p}, \mathbf{q}) = \Theta(\|\mathbf{p} - \mathbf{q}\|_2^2)$, which makes it easier to argue our algorithms are not too bad compared with the optimal one. We first consider an algorithm we call the counting algorithm $\mathcal{A}_{count}$ (Algorithm 6), which is a DOMINATION analogue of the counting algorithm proposed by [177]. We show that $\mathcal{A}_{count}$ has a competitive ratio of $\tilde{\Theta}(n)$. Intuitively, the main reason $\mathcal{A}_{count}$ fails is that $\mathcal{A}_{count}$ tries to consider samples from different coordinates equally important even when they are sampled from a very unbalanced distribution (for example, $p_1 \neq q_1, p_2 = q_2, ..., p_n = q_n$). We then consider another algorithm we call the max algorithm $\mathcal{A}_{max}$ (Algorithm 7) which simply finds $i' = \max_i |\sum_{j=1}^{r}(X_{i,j} - Y_{i,j})|$ and outputs B according the sign of $\sum_{j=1}^{r}(X_{i',j} - Y_{i',j})$. We show $\mathcal{A}_{max}$ also has a competitive ratio of $\tilde{\Theta}(n)$. Interestingly, $\mathcal{A}_{max}$ fails for a different reason from $\mathcal{A}_{count}$, namely that $\mathcal{A}_{max}$ does not use the information fully from all coordinates when the samples are sampled from a very balanced distribution. In fact, $\mathcal{A}_{count}$ performs well whenever $\mathcal{A}_{max}$ fails and vice versa. We therefore show how combine $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ in two different ways to get two new algorithms: $\mathcal{A}_{comb}$ (Algorithm 8) and $\mathcal{A}_{cube}$ (Algorithm 9). We show that both of these new algorithms have a competitive ratio of $\tilde{O}(\sqrt{n})$, which is tight by Theorem 7.8.2. While $\mathcal{A}_{cube}$ has a slightly better competitive ratio ($O(\sqrt{n})$ versus $O(\sqrt{n \log n})$), the method introduced in $\mathcal{A}_{comb}$ is more general and allows one to combine any two algorithms for DOMINATION and to obtain the better one of the two performances on any instance.

In Section 7.6, we extend $\mathcal{A}_{comb}$ to design an efficient algorithm for DOMINATION in the general regime. In this regime, $\mathbb{I}(\mathbf{p}, \mathbf{q})$ can be much larger than $\|\mathbf{p} - \mathbf{q}\|_2^2$, particularly for values of p_i and q_i very close to 0 or 1. In these corner cases, the counting algorithm $\mathcal{A}_{count}$ and max algorithm $\mathcal{A}_{max}$ can fail very badly; we will show that even for fixed n, their competitive ratios can grow arbitrarily large (Lemma 7.6.4 and Lemma 7.6.5). One main reason for this failure is that, even when $|p_i - q_i| < |p_j - q_j|$, samples from coordinate i could convey much more information than the samples from coordinate j (consider, for example, $p_i = \varepsilon/2, q_i = 0$, and $p_j = 1/2 + \varepsilon, q_j = 1/2$). Taking this into account, we design a new algorithm $\mathcal{A}_{coup}$ (Algorithm 10) which has a competitive ratio of $O(\sqrt{n} \log n)$ in the general regime. The new algorithm builds off $\mathcal{A}_{coup}$ and still combines features from both $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$, but also better estimates the importance of each coordinate. To estimate how much information each coordinate has, the new algorithm divides the samples into $\Theta(\log n)$ groups and checks how often samples from coordinate i are consistent with themselves. If one coordinate has a large proportion of the total information, it uses samples from that coordinate to decide B, otherwise it takes a majority vote on samples from all coordinates.

In Section 7.7, we return to Top-k and present an algorithm that has a competitive ratio of $\tilde{O}(\sqrt{n})$, thus proving Theorem 7.3.1. Our algorithm works by reducing the Top-k problem to several instances of the Domination problem (see Theorem 4.4.1). At a high level, the algorithm tries to find the top k rows by pairwise comparisons of rows, each of which can be thought of as an instance of Domination. We use algorithm $\mathcal{A}_{coup}$ to solve these Domination instances. Since we only need to make at most n^2 comparisons, if $\mathcal{A}_{coup}$ outputs the correct answer with at least $1 - \frac{\varepsilon}{n^2}$ probability for each comparison, then by union bound all the comparisons will be correct with probability at least $1 - \varepsilon$. However, to find the top k rows, we do not actually need to compare all the rows to each other; Lemma 7.7.1 shows that we can find the top k rows with high probability while making only $O(n)$ comparisons. Using this lemma, we get a linear time algorithm (linear in the size of the input, i.e. $\Theta(n^2 r)$) for solving Top-k. Finally in Lemma 7.7.3, we extend the lower bound for Domination proved in Lemma 7.4.1 to show a lower bound on the number of samples any algorithm would need on a specific instance of Top-k. Combining these results, we prove Theorem 7.3.1.

Finally, in Section 7.8, we show that the algorithms for both Domination and Top-k presented in the previous sections have the optimal competitive ratio (up to polylogarithmic factors). Specifically, we show that for any algorithm A solving Domination, there exists an instance C of domination where $r_{min}(C, A) \geq \tilde{\Omega}(\sqrt{n}) r_{min}(C)$ (Theorem 7.8.2). We accomplish this by constructing a distribution $\mathcal{C}$ over instances of Domination such that each instance in the support of this distribution can by solved by an algorithm with low sample complexity (Theorem 7.8.3) but any algorithm that succeeds over the entire distribution requires $\tilde{\Omega}(\sqrt{n})$ times more samples (Theorem 7.8.4). We then embed Domination in Top-k (similarly as in Section 7.7) to show an analogous $\tilde{\Omega}(\sqrt{n})$ lower bound for Top-k (Theorem 7.8.1).

7.4 Lower Bounds on the Sample Complexity of Domination

We start by establishing lower bounds on the number of samples $r_{min}(C)$ needed by any algorithm to succeed with constant probability on a given instance $C = (n, \mathbf{p}, \mathbf{q})$ of DOMINATION. This is controlled by the quantity $\mathbb{I}(\mathbf{p}, \mathbf{q})$, which is the amount of information we can learn about the bit B given one sample of pairwise comparison between each of the coordinates of $\mathbf{p}$ and $\mathbf{q}$.

Definition 7.4.1. *Given* $0 \le p, q \le 1$*, define*

$$\mathbb{I}(p, q) = (p(1-q) + q(1-p))\left(1 - H\left(\frac{p(1-q)}{p(1-q) + q(1-p)}\right)\right).$$

Given $\mathbf{p} = (p_1, \cdots, p_n) \in [0, 1]^n, \mathbf{q} = (q_1, \cdots, q_n) \in [0, 1]^n$*, define*

$$\mathbb{I}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \mathbb{I}(p_i, q_i).$$

Lemma 7.4.1. *Let* $C = (n, \mathbf{p}, \mathbf{q})$ *be an instance of* DOMINATION. *Then* $r_{min}(C) \ge 0.05/\mathbb{I}(\mathbf{p}, \mathbf{q})$.

Proof. The main idea is to bound the mutual information between the samples and the correct output, and then apply Fano's inequality. Let $\mathbf{p} = (p_1, \cdots, p_n)$ and $\mathbf{q} = (q_1, \cdots, q_n)$. Recall that B indicates the correct output and that $X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}$ are the samples given to the algorithm. By Fact 1.2.2,

$$I(B; X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) = I(B; X_{1,1}Y_{1,1}) + I(B; X_{1,2}, ..., X_{n,r}, Y_{1,2}, ..., Y_{n,r} | X_{1,1}Y_{1,1}).$$

When $\mathbf{p}$, $\mathbf{q}$ and B are given, each sample ($X_{i,j}$ or $Y_{i,j}$) is independent of the other samples, and thus $I(X_{1,1}Y_{1,1}; X_{1,2}, ..., X_{n,r}, Y_{1,2}, ..., Y_{n,r}|B) = 0$. By Fact 1.2.3, we then have

$$I(B; X_{1,2}, ..., X_{n,r}, Y_{1,2}, ..., Y_{n,r}|X_{1,1}Y_{1,1}) \leq I(B; X_{1,2}, ..., X_{n,r}, Y_{1,2}, ..., Y_{n,r})$$

and therefore

$$I(B; X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) \leq I(B; X_{1,1}Y_{1,1}) + I(B; X_{1,2}, ..., X_{n,r}, Y_{1,2}, ..., Y_{n,r}).$$

Repeating this, we get

$$I(B; X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) \leq \sum_{i=1}^{n}\sum_{j=1}^{r} I(B; X_{i,j}Y_{i,j}).$$

By Fact 1.2.5, we have

$$\begin{aligned}
& I(B; X_{i,j}Y_{i,j}) \\
= \;& \Pr[B=0] \cdot D(X_{i,j}Y_{i,j}|B=0 \| X_{i,j}Y_{i,j}) + \Pr[B=1] \cdot D(X_{i,j}Y_{i,j}|B=1 \| X_{i,j}Y_{i,j}) \\
= \;& (p_i(1-q_i) + q_i(1-p_i)) \left(1 - H\left(\frac{p_i(1-q_i)}{p_i(1-q_i)+q_i(1-p_i)}\right)\right) \\
= \;& \mathbb{I}(p_i, q_i).
\end{aligned}$$

It follows that

$$I(B; X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) \leq \sum_{i=1}^{n}\sum_{j=1}^{r} I(B; X_{i,j}Y_{i,j}) = r \cdot \sum_{i=1}^{n} \mathbb{I}(p_i, q_i) = r\mathbb{I}(\mathbf{p}, \mathbf{q}).$$

For any algorithm, let p_e be its error probability on DOMINATION$(n, \mathbf{p}, \mathbf{q}, r)$. By Fano's inequality, we have that

$$\begin{aligned} H(p_e) &\geq H(B|X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) \\ &= H(B) - I(B; X_{1,1}, X_{1,2}, ..., X_{n,r}, Y_{1,1}, ..., Y_{n,r}) \\ &= 1 - r\mathbb{I}(\mathbf{p}, \mathbf{q}) \geq 0.95. \end{aligned}$$

Since $H(p_e) \geq 0.95$, we find that $p_e \geq 1/4$, as desired. □

In the following section, we will concern ourselves with instances $C = (n, \mathbf{p}, \mathbf{q})$ that satisfy $\delta \leq p_i, q_i \leq 1-\delta$ for some constant δ for all i. For such instances, we can approximate $\mathbb{I}(p, q)$ by the ℓ_2 distance between $\mathbf{p}$ and $\mathbf{q}$.

Lemma 7.4.2. *For some $0 < \delta \leq \frac{1}{2}$, let $\delta \leq p, q \leq 1 - \delta$. Then*

$$\frac{1}{4 \ln 2}(p - q)^2 \leq \mathbb{I}(p, q) \leq \frac{1}{\delta \ln 2}(p - q)^2.$$

Proof. Let $x = p(1-q)$ and $y = q(1-p)$. Then $\mathbb{I}(p, q) = (x+y)(1-H(\frac{x}{x+y}))$ and $p-q = x-y$. We need to show that

$$(x + y)\left(1 - H\left(\frac{x}{x + y}\right)\right) \leq \frac{1}{\delta \ln 2}(x - y)^2.$$

By Fact 1.2.6,

$$\frac{1}{\ln 2}z^2 \leq 1 - H\left(\frac{1}{2} + z\right) = D\left(\frac{1}{2} + z \middle\| \frac{1}{2}\right) \leq \frac{4}{\ln 2}z^2,$$

and therefore

$$\frac{1}{4 \ln 2}\frac{(x - y)^2}{(x + y)} \leq (x + y)\left(1 - H\left(\frac{x}{x + y}\right)\right) \leq \frac{1}{\ln 2}\frac{(x - y)^2}{(x + y)}.$$

Since

$$x + y = p(1-q) + q(1-p) \geq 2\sqrt{p(1-p)q(1-q)} \geq 2\delta(1-\delta) \geq \delta,$$

this implies the desired upper bound. The lower bound also holds since,

$$x + y = p(1-q) + q(1-p) \leq \sqrt{p^2 + (1-p)^2} \cdot \sqrt{q^2 + (1-q)^2} \leq \delta^2 + (1-\delta)^2 \leq 1.$$

□

Corollary 7.4.1. *Let $C = (n, \mathbf{p}, \mathbf{q})$ be an instance of* DOMINATION *satisfying $\delta \leq \mathbf{p}_i, \mathbf{q}_i \leq 1 - \delta$ for all $i \in [n]$. Then*

$$r_{min}(C) \geq 0.05 \ln(2) \cdot \frac{\delta}{\|\mathbf{p} - \mathbf{q}\|_2^2}.$$

Proof. By Lemma 7.4.2, $\mathbb{I}(\mathbf{p}, \mathbf{q}) \leq \|\mathbf{p} - \mathbf{q}\|_2^2/(\delta \ln 2)$. The result then follows from Lemma 7.4.1. □

7.5 Domination in the Well-behaved Regime

We now proceed to the problem of designing algorithms for DOMINATION which are competitive on all instances. As a warmup, we begin by considering only instances $C = (n, \mathbf{p}, \mathbf{q})$ of DOMINATION satisfying $\delta \leq p_i, q_i \leq 1 - \delta$ for all $i \in [n]$ where $0 < \delta \leq 1/2$ is some fixed constant. This regime of instances captures much of the interesting behavior of DOMINATION, but with the added benefit that the mutual information between the samples and B behaves nicely in this regime: in particular $\mathbb{I}(\mathbf{p}, \mathbf{q}) = \Theta(\|\mathbf{p} - \mathbf{q}\|_2^2)$ (see Lemma 7.4.2). By Corollary 7.4.1, we have $r_{min} \geq \Omega(\frac{1}{\|\mathbf{p}-\mathbf{q}\|_2^2})$. This fact will make it easier to design algorithms for DOMINATION which are competitive in this regime.

In Section 7.5.1, we give two simple algorithms (counting algorithm and max algorithm) which can solve DOMINATION given $\tilde{O}(n/\|\mathbf{p} - \mathbf{q}\|_2^2)$ samples which gives them a competitive ratio of $\tilde{O}(n)$. We will then show that this is tight, i.e. their competitive ratio is $\tilde{\Theta}(n)$ in Lemma 7.5.3 and Lemma 7.5.4. While the sample complexities of these two algorithms are not optimal, they have the nice property that whenever one performs badly, the other performs well. In Section 7.5.2, we show how to combine the counting algorithm and the max algorithm to give two different algorithms which can solve DOMINATION using only $\tilde{O}(\sqrt{n}/\|\mathbf{p}-\mathbf{q}\|_2^2)$ samples i.e. they have a competitive ratio of $\tilde{O}(\sqrt{n})$. According to Theorem 7.8.2, this is the best we can do up to polylogarithmic factors.

7.5.1 Counting algorithm and max algorithm

We now consider two simple algorithms for DOMINATION$(n, \mathbf{p}, \mathbf{q})$, which we call the *counting algorithm* (Algorithm 6) and the *max algorithm* (Algorithm 7) denoted by $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ respectively. We show that both algorithms require $\tilde{O}(\frac{n}{\|\mathbf{p}-\mathbf{q}\|_2^2})$ samples to solve DOMINATION (Lemmas 7.5.1 and 7.5.2). By Corollary 7.4.1, we have $r_{min} \geq \Omega(\frac{1}{\|\mathbf{p}-\mathbf{q}\|_2^2})$, leading to a $\tilde{O}(n)$ competitive ratio for these algorithms. We show in Lemma 7.5.3 and Lemma 7.5.4 that this is tight up to polylogarithmic factors i.e. their competitive ratio is $\tilde{\Theta}(n)$.

Algorithm 6 The counting algorithm $\mathcal{A}_{count}$ for DOMINATION$(n, \mathbf{p}, \mathbf{q}, r)$

for $i = 1$ to n **do**
  $S_i = \sum_{j=1}^{r}(X_{i,j} - Y_{i,j})$
end for
$Z = \sum_{i=1}^{n} S_i$
If $Z > 0$, output $B = 0$. If $Z < 0$, output $B = 1$. If $Z = 0$, output $B = 0$ with probability $1/2$ and output $B = 1$ with probability $1/2$.

Algorithm 7 The max algorithm $\mathcal{A}_{max}$ for DOMINATION$(n, \mathbf{p}, \mathbf{q}, r)$

for $i = 1$ to n **do**
  $S_i = \sum_{j=1}^{r}(X_{i,j} - Y_{i,j})$
end for
$i' = \arg\max |S_i|$
$Z = S_{i'}$
If $Z > 0$, output $B = 0$. If $Z < 0$, output $B = 1$. If $Z = 0$, output $B = 0$ with probability $1/2$ and output $B = 1$ with probability $1/2$.

Both the counting algorithm and the max algorithm begin by computing (for each coordinate i) the differences between the number of ones in the $X_{i,j}$ samples and $Y_{i,j}$ samples; i.e., we compute the values $S_i = \sum_{j=1}^{r}(X_{i,j} - Y_{i,j})$. The counting algorithm $\mathcal{A}_{count}$ decides whether to output $B = 0$ or $B = 1$ based on the sign of $\sum_i S_i$, whereas the max algorithm decides its output based on the sign of the S_i with the largest absolute value. See Algorithms 6 and 7 for detailed pseudocode for both $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$.

We omit proofs in this subsection. They can be found in the full version of [57].

We begin by proving upper bounds for the sample complexities of both $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$. In particular, both $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ need at most $\tilde{O}(n)$ times as many samples as the best possible algorithm for any instance in this regime.

Lemma 7.5.1. *Let $C = (n, \mathbf{p}, \mathbf{q})$ be an instance of* Domination. *Then*

$$r_{min}(C, \mathcal{A}_{count}, 1-\alpha) \leq \frac{2n\ln(\alpha^{-1})}{\|\mathbf{p}-\mathbf{q}\|_1^2}.$$

If C further satisfies $\delta \leq p_i, q_i \leq 1-\delta$ for all i for some constant $\delta > 0$, then

$$r_{min}(C, \mathcal{A}_{comb}) \leq O(n) r_{min}(C).$$

Lemma 7.5.2. *Let $C = (n, \mathbf{p}, \mathbf{q})$ be an instance of* Domination. *Then*

$$r_{min}(C, \mathcal{A}_{max}, 1-\alpha) \leq \frac{8\ln(2n\alpha^{-1})}{\|\mathbf{p}-\mathbf{q}\|_\infty^2}$$

If C further satisfies $\delta \leq p_i, q_i \leq 1-\delta$ for all i for some constant δ, then

$$r_{min}(C, \mathcal{A}_{comb}) \leq O(n\log n) r_{min}(C).$$

We now show that the upper bounds we proved above are essentially tight. In particular, we demonstrate instances where both $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ need $\tilde{\Omega}(n)$ times as many samples as the best possible algorithms for those instances. Interestingly, on the instance where $\mathcal{A}_{count}$ suffers, $\mathcal{A}_{max}$ performs well, and vice versa. This fact will prove useful in the next section.

Lemma 7.5.3. *For each $\varepsilon < \frac{1}{10}$ and each sufficiently large n, there exists an instance $C = (n, \mathbf{p}, \mathbf{q})$ of* Domination *such that the following two statements are true:*

1. $r_{min}(C, \mathcal{A}_{max}, 1 - \frac{2}{n}) \le \frac{16 \ln n}{\varepsilon^2}$.

2. $r_{min}(C, \mathcal{A}_{count}) \ge \frac{n}{128\varepsilon^2}$.

It is not hard to observe that in certain cases, the counting algorithm of [177] for Top-k reduces to the algorithm $\mathcal{A}_{count}$ for Domination. It follows that there also exists an $\Omega(n)$ multiplicative gap between the sample complexity of their counting algorithm and the sample complexity of the best algorithm on some instances.

Corollary 7.5.1. *Let A' be the* Top-k *algorithm of [177], and let $S = (n, k, \mathbf{P})$ be a* Top-k *instance, with $\mathbf{P}$ as described in Section 7.3. Then, for sufficiently large n and $\varepsilon < 1/10$, $r_{min}(S, A') \ge \Omega(\frac{n}{\varepsilon^2})$.*

We will now show that $\mathcal{A}_{max}$ has a competitive ratio of $\tilde{\Omega}(n)$.

Lemma 7.5.4. *For each sufficiently large n, there exists an instance $C = (n, \mathbf{p}, \mathbf{q})$ of* Domination *such that the following two statements are true:*

1. $r_{min}(C, \mathcal{A}_{count}, 1 - \frac{1}{n}) \le 2n^3 \ln n$.

2. $r_{min}(C, \mathcal{A}_{max}, \frac{4}{5}) \ge \frac{n^4}{2^{14} \ln n}$.

7.5.2 $\tilde{O}(\sqrt{n})$-competitive algorithms

We will now demonstrate two algorithms for Domination that use at most $\tilde{O}(\sqrt{n})$ times more samples than the best possible algorithm for each instance. According to Theorem 7.8.2, this is the best we can do up to polylogarithmic factors.

Note that the counting algorithm $\mathcal{A}_{count}$ tends to work well when the max algorithm $\mathcal{A}_{max}$ fails, and vice versa (e.g., Lemmas 7.5.3 and 7.5.4). Therefore, intuitively, combining both algorithms in some way should lead to better performance.

Both of the algorithms we present in this section share this intuition. We begin (in Lemma 7.5.5) by demonstrating a very general method for combining any two algorithms for DOMINATION. Applying this to $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$, we obtain an algorithm $\mathcal{A}_{comb}$ that satisfies $r_{min}(C, \mathcal{A}_{comb}) \leq O(\sqrt{n \log n}) \cdot r_{min}(C)$ (Corollary 7.5.2) for instances C in this regime. We then show an alternate algorithm with slightly better performance than $\mathcal{A}_{comb}$, which we call the *sum of cubes* algorithm $\mathcal{A}_{cube}$. This algorithm satisfies $r_{min}(C, \mathcal{A}_{cube}) \leq O(\sqrt{n}) \cdot r_{min}(C)$ for instances C in this regime (Theorem 7.5.1).

Combining counting and max

We first show how to combine any two algorithms for DOMINATION to get an algorithm that always does at least as well as the better of the two algorithms. Call an algorithm $\mathcal{A}$ for DOMINATION *stable* if it always outputs the correct answer with probability at least 1/2 (i.e. it always does at least as well as a random guess). Note that $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ are both stable. We have the following lemma.

Lemma 7.5.5. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two stable algorithms for* DOMINATION. *Then there exists an algorithm $\mathcal{A}_{comb}$ such that for all instances C of* DOMINATION,

$$r_{min}(C, \mathcal{A}_{comb}, 1 - \alpha) \leq 32 \ln(\alpha^{-1}) \cdot \min\left(r_{min}(C, \mathcal{A}_1), r_{min}(C, \mathcal{A}_2)\right)$$

Proof. See Algorithm 8 for a description of $\mathcal{A}_{comb}$. Assume without loss of generality that $B = 0$, and let $r = 32 \log(n\alpha^{-1}) \min\left(r_{min}(C, \mathcal{A}_1), r_{min}(C, \mathcal{A}_2)\right)$. We will show that $\mathcal{A}_{comb}$ outputs $B = 0$ correctly with probability at least $1 - \alpha$.

Let $r' = \frac{r}{32 \ln n}$; note that either $r' \geq r_{min}(C, \mathcal{A}_1)$ or $r_{min}(C, \mathcal{A}_2)$. Assume first that $r' \geq r_{min}(C, \mathcal{A}_1)$. Then, $\mathcal{A}_1$ will output $B = 0$ in each of its $16 \ln \alpha^{-1}$ groups with probability at least $\frac{3}{4}$. On the other hand, since it is stable, $\mathcal{A}_2$ will output $B = 0$ in each of its groups with probability at least $\frac{1}{2}$. Therefore

$$\mathrm{E}\left[\frac{Z_1 + Z_2}{2}\right] \leq \frac{1}{8} + \frac{1}{4} \leq \frac{3}{8}.$$

Since $\frac{Z_1+Z_2}{2}$ is the average of $32 \ln \alpha^{-1}$ random variables, by Hoeffding's inequality, the probability that $\frac{Z_1+Z_2}{2} \geq \frac{1}{2}$ is at most $\exp\left(-2(32 \ln \alpha^{-1})(\frac{1}{8})^2\right) \leq \alpha$.

Similarly, if $r' \geq r_{min}(C, \mathcal{A}_2)$, the probability that $\frac{Z_1+Z_2}{2} \geq \frac{1}{2}$ is also at most α. This concludes the proof. □

Algorithm 8 Combining two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ for Domination$(n, \mathbf{p}, \mathbf{q}, r)$

Divide the samples into $32 \ln \alpha^{-1}$ groups.
Run $\mathcal{A}_1$ on each of the first $16 \ln \alpha^{-1}$ groups and let Z_1 be the average of the outputs.
Run $\mathcal{A}_2$ on each of the last $16 \ln \alpha^{-1}$ groups and let Z_2 be the average of the outputs.
If $\frac{Z_1+Z_2}{2} \leq \frac{1}{2}$ output $B = 0$, else output $B = 1$.

Corollary 7.5.2. *Let $\mathcal{A}_{comb}$ be the algorithm we obtain by combining $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ in the manner of Lemma 7.5.5. Then for any instance $C = (n, \mathbf{p}, \mathbf{q})$ of* Domination*,*

$$r_{min}(C, \mathcal{A}_{comb}) \leq O\left(\frac{\sqrt{n \log n}}{\|\mathbf{p} - \mathbf{q}\|_2^2}\right).$$

If C further satisfies $\delta \leq p_i, q_i \leq 1 - \delta$ for all i for some constant δ, then

$$r_{min}(C, \mathcal{A}_{comb}) \leq O(\sqrt{n \log n}) r_{min}(C).$$

Proof. This follows from Lemmas 7.5.1, 7.5.2, 7.5.5, and the following observation:

$$\begin{aligned} \min\left(\frac{n}{\|\mathbf{p} - \mathbf{q}\|_1^2}, \frac{\log n}{\|\mathbf{p} - \mathbf{q}\|_\infty^2}\right) &\leq \sqrt{\frac{n}{\|\mathbf{p} - \mathbf{q}\|_1^2} \cdot \frac{\log n}{\|\mathbf{p} - \mathbf{q}\|_\infty^2}} \\ &\leq \frac{\sqrt{n \log n}}{\|\mathbf{p} - \mathbf{q}\|_2^2}. \end{aligned}$$

The last inequality follows from the fact that for any vector $\mathbf{x}$, $\|\mathbf{x}\|_2^2 \le \|\mathbf{x}\|_1 \cdot \|\mathbf{x}\|_\infty$. The second part of the corollary then follows directly from Corollary 7.4.1. $\square$

The sum of cubes algorithm

We now give a different algorithm for Domination which we call the *sum of cubes algorithm*, $\mathcal{A}_{cube}$. If we let $S_i = \sum_j (X_i - Y_i)$, then intuitively, whereas $\mathcal{A}_{count}$ decides its output based on the signed ℓ_1 norm of the S_i and whereas $\mathcal{A}_{max}$ decides its output based on the signed ℓ_∞ norm of the S_i, $\mathcal{A}_{cube}$ decides its output based on the signed ℓ_3 norm of the S_i. See Algorithm 9 for a detailed description of the algorithm.

Algorithm 9 Sum of cubes algorithm $\mathcal{A}_{cube}$ for Domination$(n, \mathbf{p}, \mathbf{q}, r)$

1: $T_{i,j} = 1$ with probability $\frac{1}{2} + \frac{(X_{i,j} - Y_{i,j})}{2}$ and $T_{i,j} = -1$ with probability $\frac{1}{2} - \frac{(X_{i,j} - Y_{i,j})}{2}$
2: $S_i = \sum_{j=1}^r T_{i,j}$
3: $Z = \sum_{i=1}^n S_i^3$
4: If $Z \ge 0$, output $B = 0$. If $Z < 0$, output $B = 1$.

To analyze the performance of $\mathcal{A}_{cube}$, we begin by analyzing statistical properties of the random variable S.

Lemma 7.5.6. *Let $S = \sum_{j=1}^r X_j$ where $X_1, \cdots, X_r$ are i.i.d $\{-1, 1\}$-valued random variables with mean $\varepsilon \ge 0$ and $r \ge 8$. Let $Z = S^3$. Then*

$$\mathrm{E}[Z] \ge 2r^2\varepsilon + \frac{1}{2}r^3\varepsilon^3$$
$$\mathrm{Var}[Z] \le 15r^3 + 36r^4\varepsilon^2 + 9r^5\varepsilon^4.$$

Proof. By applying the multinomial theorem and using the fact that $X_i^2 = 1$ for each i, we can write multilinear expressions for S^3 and S^6. We can now use linearity of expectation and the independence among the X_i's to compute the mean and variance exactly.

$$\begin{aligned}
\mathrm{E}[Z] &= \mathrm{E}[S^3] = (-2r + 3r^2)\varepsilon + (2r - 3r^2 + r^3)\varepsilon^3 \geq 2r^2\varepsilon + \frac{1}{2}r^3\varepsilon^3 \\
\mathrm{Var}[Z] &= E[S^6] - \mathrm{E}[S^3]^2 = (16r - 30r^2 + 15r^3) + (-136r + 282r^2 - 183r^3 + 36r^4)\varepsilon^2 + \\
&\qquad (240r - 522r^2 + 381r^3 - 108r^4 + 9r^5)\varepsilon^4 + (-120r + 270r^2 - 213r^3 + 72r^4 - 9r^5)\varepsilon^6 \\
&\leq 15r^3 + 36r^4\varepsilon^2 + 9r^5\varepsilon^4
\end{aligned}$$

□

Lemma 7.5.7. *Let* $S_i = \sum_{j=1}^{r} X_{i,j}$ *where for each* $i \in [n]$, $X_{i,1}, \cdots, X_{i,r}$ *are i.i.d* $\{-1, 1\}$*-valued random variables with mean* ε_i*, along with the condition that either all* $\varepsilon_i \geq 0$ *or all* $\varepsilon_i \leq 0$*. Let* $Z = \sum_{i=1}^{n} S_i^3$*. If* $r \geq 8$ *and* $r \geq \eta\sqrt{n}/(\sum_{i=1}^{n} \varepsilon_i^2)$ *for some* $\eta \geq 1$ *then,* $\mathrm{E}[Z]^2 \geq \frac{\eta}{36}\mathrm{Var}[Z]$.

Proof. Without loss of generality, we can assume that $\varepsilon_i \geq 0$ for all $i \in [n]$. By Lemma 7.5.6,

$$\mathrm{E}[Z]^2 \geq 4r^4(\sum_i \varepsilon_i)^2 + \frac{1}{4}r^6(\sum_i \varepsilon_i^3)^2 + 2r^5(\sum_i \varepsilon_i)(\sum_i \varepsilon_i^3) \tag{7.2}$$

$$\mathrm{Var}[Z] \leq 15nr^3 + 36r^4\sum_i \varepsilon_i^2 + 9r^5\sum_i \varepsilon_i^4. \tag{7.3}$$

We will show that each term in the Equation 7.3 is dominated by some term in Equation 7.2.

$$nr^3 = r^5 \frac{n}{r^2} \leq \frac{1}{\eta^2} r^5 (\sum_i \varepsilon_i^2)^2 \leq \frac{1}{\eta^2} r^5 (\sum_i \varepsilon_i)(\sum_i \varepsilon_i^3) \qquad \text{(Cauchy-Schwarz inequality)}$$

$$r^4 (\sum_i \varepsilon_i^2) \leq \frac{1}{\eta\sqrt{n}} r^5 (\sum_i \varepsilon_i^2)^2 \leq \frac{1}{\eta\sqrt{n}} r^5 (\sum_i \varepsilon_i)(\sum_i \varepsilon_i^3)$$

$$r^5 (\sum_i \varepsilon_i^4) \leq r^6 \frac{1}{\eta\sqrt{n}} (\sum_i \varepsilon_i^2)(\sum_i \varepsilon_i^4) \leq r^6 \frac{1}{\eta\sqrt{n}} \left(\sqrt{n} \cdot (\sum_i \varepsilon_i^4)^{1/2} \right) (\sum_i \varepsilon_i^4)$$

$$\text{(Cauchy-Schwarz inequality)}$$

$$= \frac{r^6}{\eta} (\sum_i \varepsilon_i^4)^{3/2} \leq \frac{r^6}{\eta} (\sum_i \varepsilon_i^3)^2 \quad \text{(monotonicity of } \ell_p \text{ norms)}$$

Adding the above inequalities, we get $\mathrm{Var}[Z] \leq \frac{36}{\eta} \mathrm{E}[Z]^2$. □

Theorem 7.5.1. *If $C = (n, \mathbf{p}, \mathbf{q})$ is any instance of* DOMINATION*, then*

$$r_{min}(C, \mathcal{A}_{cube}) \leq \max \left(\frac{144\sqrt{n}}{\|\mathbf{p} - \mathbf{q}\|_2^2}, 8 \right).$$

If C satisfies $\delta \leq p_i, q_i \leq 1 - \delta$ for all i for some constant δ, then

$$r_{min}(C, \mathcal{A}_{cube}) \leq O(\sqrt{n}) r_{min}(C).$$

Proof. Assume without loss of generality that $B = 0$. We have $S_i = \sum_{j=1}^{r} T_{i,j}$ and $Z = \sum_{i=1}^{n} S_i^3$. Note that for each i, the $T_{i,j}$ are i.i.d. $\{-1, 1\}$ random variables with mean $\mathrm{E}[T_{i,j}] = p_i - q_i$. Applying Lemma 7.5.7 with $\eta = 144$, if $r \geq \max \left(\frac{144\sqrt{n}}{\|\mathbf{p}-\mathbf{q}\|_2^2}, 8 \right)$ we have that $\mathrm{E}[Z]^2 \geq \frac{144}{36} \mathrm{Var}[Z] = 4\mathrm{Var}[Z]$. Since the algorithm makes an error (i.e. outputs $B = 1$) when $Z < 0$, we can use Chebyshev's inequality to bound the probability that $Z < 0$.

$$\Pr[Z < 0] \leq \Pr[|Z - \mathrm{E}[Z]| \geq \mathrm{E}[Z]] \leq \frac{\mathrm{Var}[Z]}{\mathrm{E}[Z]^2} \leq \frac{1}{4}.$$

The second part of the theorem then follows directly from Corollary 7.4.1. □

7.6 Domination in the General Regime

In this section, we consider DOMINATION in the general regime. Unlike in the previous section, it is no longer true that $I(X_{i,j}Y_{i,j}; B) = \mathbb{I}(p_i, q_i) = \Theta((p_i - q_i)^2)$. In particular, when p_i and q_i are both very small, $\mathbb{I}(p_i, q_i)$ can be much bigger than $(p_i - q_i)^2$; as a result, the algorithms designed in the previous section can fail under these circumstances.

In Section 7.6.1, we present a new algorithm $\mathcal{A}_{coup}$ which is $\tilde{O}(\sqrt{n} \cdot r_{min})$-competitive. According to Theorem 7.8.2, this is the best we can do up to polylogarithmic factors. In Section 7.6.2, we then demonstrate that the general regime is indeed harder than the restricted regime in Section 7.5. In particular, we give instances where the algorithms presented in the previous section fail; we show that the competitive ratio of these algorithms is unbounded (even for fixed n).

7.6.1 An $\tilde{O}(\sqrt{n})$-competitive algorithm

Here we give an algorithm that only needs $O(\sqrt{n}\log(n)/\mathbb{I}(\mathbf{p}, \mathbf{q}))$ samples to solve DOMINATION (Theorem 7.6.1). By Lemma 7.4.1, this is only $\tilde{O}(\sqrt{n})$ times as many samples as the optimal algorithm needs. Intuitively, the algorithm works as follows: if for some coordinate i, $X_{i,1}Y_{i,1}...X_{i,r}, Y_{i,r}$ conveys enough information about B, we will only use samples from coordinate i to determine B. Otherwise, the information about B must be well-spread throughout all the coordinates, and a majority vote will work.

We begin by bounding the probability we can determine the answer from a single fixed coordinate. The following four lemmas will be used to prove Theorem 7.6.1 and their proofs can be found in the full version of [57].

Lemma 7.6.1 (Sanov's theorem)**.** *Let $\mathcal{P}(\Sigma)$ denote the space of all probability distributions on some finite set Σ. Let $R \in \mathcal{P}(\Sigma)$ and let $Z_1, \cdots, Z_k$ be i.i.d random variables with distribution R. For every $x \in \Sigma^k$, we can define an empirical probability distribution $\hat{P}_x$ on*

Σ *as*

$$\forall \sigma \in \Sigma \quad \hat{P}_x(\sigma) = \frac{|\{i \in [k] : x_i = \sigma\}|}{k}.$$

Let C be a closed convex subset of $\mathcal{P}(\Sigma)$ such that for some $P \in C$, $D(P||R) < \infty$. Then

$$\Pr\left[\hat{P}_{(Z_1,\cdots,Z_k)} \in C\right] \leq \exp\left(-k(\ln 2) D(Q^*||R)\right)$$

where $Q^* = \operatorname{argmin}_{Q \in C} D(Q||R)$ is unique. In the case when $D(Q||R) = \infty$ for all $Q \in C$, $\Pr\left[\hat{P}_{(Z_1,\cdots,Z_k)} \in C\right] = 0$.

Proof. See exercise 2.7 and 3.20 in [66]. □

Sanov's theorem allows us to bound the following probability that we incorrectly rank two Bernoulli variables (e.g., X_i and Y_i for a fixed coordinate i) from k independent samples.

Lemma 7.6.2. *Let $0 \leq q < p \leq 1$ and let $X_1, \cdots, X_k$ be i.i.d $\mathcal{B}(p)$ and $Y_1, \cdots, Y_k$ be i.i.d $\mathcal{B}(q)$. Then*

$$\Pr\left[\sum_{i=1}^{k}(X_i - Y_i) \leq 0\right] \leq \exp\left(-2(\ln 2) k \log\left(\frac{1}{\sqrt{pq} + \sqrt{(1-p)(1-q)}}\right)\right).$$

We can in turn relate the upper bound in Lemma 7.6.2 to the quantity $\mathbb{I}(p,q)$.

Lemma 7.6.3.

$$2\log\left(\frac{1}{\sqrt{pq} + \sqrt{(1-p)(1-q)}}\right) \geq \frac{1}{2}\mathbb{I}(p,q).$$

Combining Lemma 7.6.2 and Lemma 7.6.3, we can show the following corollary which says that given $\Omega(1/\mathbb{I}(p,q))$ samples, we can correctly rank two Bernoulli variables with constant probability.

Corollary 7.6.1. *In* Domination$(n, \mathbf{p}, \mathbf{q}, r)$*, for any $i \in [n]$, if $r > 6/\mathbb{I}(p_i, q_i)$, then*

$$\Pr\left[\operatorname{sign}\left(\sum_{j=1}^{r}(X_{i,j} - Y_{i,j})\right) = (-1)^B\right] > 5/6.$$

Proof. Assume we are in the $B = 0$ case, the other case is similar. Fix an $i \in [n]$. By Lemma 7.6.2,

$$\Pr\left[\sum_{j=1}^{r}(X_{i,j} - Y_{i,j}) \le 0\right] \le \exp\left(-r(\ln 2)\log\left(\frac{1}{\sqrt{p_i q_i} + \sqrt{(1-p_i)(1-q_i)}}\right)\right)$$
$$\le \exp\left(-r(\ln 2) I_i/2\right) \qquad \text{(By Lemma 7.6.3)}$$
$$= 2^{-rI_i/2} < 1/8.$$

□

We now introduce what we call the *general coupling algorithm* $\mathcal{A}_{coup}$ for DOMINATION. A detailed description of the algorithm can be found in Algorithm 10; more briefly the algorithm works as follows:

1. Split the r samples for each of the n coordinates into $\ell = 18\log(2n\alpha^{-1})$ equally-sized segments where α is the error parameter. For each coordinate i and segment j, set $S_{i,j} = 1$ if more samples from X equal 1 than samples from Y, and -1 otherwise. This can be thought of as running a miniature version of the counting algorithm on each segment; $S_{i,j} = 1$ is evidence that $B = 0$, and $S_{i,j} = -1$ is evidence that $B = -1$.

2. Let i' be the coordinate i which maximizes $\left|\sum_{j=1}^{\ell} S_{i,j}\right|$ (i.e. the coordinate that is "most consistently" either 1 or -1). If $\left|\sum_{j=1}^{\ell} S_{i',j}\right| \ge \ell/3$ (i.e. at least $2\ell/3$ of the segments for this coordinate agree on the value of B), output B according to the sign of $\sum_{j=1}^{\ell} S_{i',j}$.

3. Otherwise, for each segment, take the majority of the votes from each of the n coordinates; that is, for each $1 \le j \le \ell$, set $T_j = \text{sign}(\sum_{i=1}^{n} S_{i,j})$. Then take another majority over the segments, by setting $Z_2 = \text{sign}(\sum_{j=1}^{\ell} T_j)$. Finally, if $Z_2 > 0$ output $B = 0$; otherwise, output $B = 1$.

Algorithm 10 General coupling algorithm $\mathcal{A}_{coup}$ for DOMINATION$(n, \mathbf{p}, \mathbf{q}, r)$

$\ell = 18\log(2n\alpha^{-1})$.
for $i = 1$ to n **do**
  for $j = 1$ to ℓ **do**
    $S_{i,j} = \text{sign}(\sum_{t=(j-1)*(r/\ell)+1}^{jr/\ell} X_{i,t} - Y_{i,t})$
    If $S_{i,j} = 0$, let $S_{i,j} = 1$ with probability $1/2$ and let $S_{i,j} = -1$ with probability $1/2$.
  end for
end for
$i' = \arg\max_i |\sum_{j=1}^{\ell} S_{i,j}|$
$Z_1 = \sum_{j=1}^{\ell} S_{i',j}$
if $|Z_1| \geq \ell/3$ **then**
  If $Z_1 > 0$ output $B = 0$, else output $B = 1$.
else
  for $j = 1$ to l **do**
    $T_j = \text{sign}(\sum_{i=1}^{n} S_{i,j})$.
    If $T_j = 0$, let $T_j = 1$ with probability $1/2$ and let $T_j = -1$ with probability $1/2$.
  end for
  $Z_2 = \text{sign}(\sum_{j=1}^{\ell} T_j)$.
  If $Z_2 = 0$, let $Z_2 = 1$ with probability $1/2$ and let $Z_2 = -1$ with probability $1/2$.
  If $Z_2 > 0$ output $B = 0$, else output $B = 1$.
end if

Theorem 7.6.1. *If $C = (n, \mathbf{p}, \mathbf{q})$ is any instance of* DOMINATION, *then*

$$r_{min}(C, \mathcal{A}_{coup}, 1-\alpha) \leq \frac{2592\sqrt{n}\ln(2n\alpha^{-1})}{\mathbb{I}(\mathbf{p}, \mathbf{q})}$$

and thus

$$r_{min}(C, \mathcal{A}_{coup}) \leq O(\sqrt{n}\log n) \cdot r_{min}(C).$$

Proof. Let $I_i = \mathbb{I}(p_i, q_i)$, $r = 2592\sqrt{n}\log(2n\alpha^{-1})/\mathbb{I}(\mathbf{p}, \mathbf{q})$ and $\ell = 18\ln(2n\alpha^{-1})$. There are two cases to consider:

1. **Case 1**: There exists an i' such that $24\sqrt{n}I_{i'} \geq \sum_{k=1}^{n} I_k$.

By symmetry, we can assume that $B = 0$. In this case, we have that $\frac{r}{\ell} \geq \frac{24 \cdot 6\sqrt{n}}{\sum_{k=1}^{n} I_k} \geq \frac{6}{I_{i'}}$. By Corollary 7.6.1, for each $j = 1, \ldots, \ell$, $\Pr[S_{i',j} = 1] \geq 5/6$. Therefore we have

$$\mathrm{E}\left[\sum_{j=1}^{l} S_{i',j}\right] \geq \ell \cdot (5/6 - 1/6) = 2\ell/3.$$

Since $S_{i',1}, ..., S_{i',l}$ are independent when B is given, by the Chernoff bound, we have that

$$\Pr\left[\sum_{j=1}^{l} S_{i',j} \geq \ell/3\right] \geq 1 - \exp(-\ell \cdot (1/3)^2 \cdot (1/2)) \geq 1 - \frac{\alpha}{2n}.$$

For $i \neq i'$, since $p_i \geq q_i$, we still have $\Pr[S_{i,j} = 1] \geq 1/2$. By a similar argument, we get

$$\Pr\left[\sum_{j=1}^{l} S_{i,j} \geq -\ell/3\right] \geq 1 - \exp(-\ell \cdot (1/3)^2 \cdot (1/2)) \geq 1 - \frac{\alpha}{2n}.$$

Let W be the event that $\sum_{j=1}^{\ell} S_{i',j} \geq \ell/3$ and for $i \neq i'$, $\sum_{j=1}^{l} S_{i,j} \geq -\ell/3$. By the union bound, we have that $\Pr[W] \geq 1 - n \cdot \frac{\alpha}{2n} = 1 - \frac{\alpha}{2}$. Moreover, when W happens, we know that $Z_1 \geq \ell/3$ and $\mathcal{A}_{coup}$ outputs $B = 0$. Therefore, in Case 1, the probability that $\mathcal{A}_{coup}$ outputs B correctly is at least $1 - \frac{\alpha}{2}$.

2. **Case 2**: For all $i \in \{1, \ldots, n\}$, $24\sqrt{n} I_i < \sum_{k=1}^{n} I_k$.

   Similarly as in Case 1, since $\Pr[S_{i,j} = (-1)^B] \geq 1/2$, the probability that $|Z_1| \geq \ell/3$ and our algorithm outputs wrongly is at most $\frac{\alpha}{2}$. For the rest of Case 2, assume $|Z_1| < \ell/3$.

   Now fix a coordinate i. Our plan is to first lower bound the amount of information samples from coordinate i have about B by using Corollary 7.6.1 and the subadditivity of information. Let $s = r/\ell$, and let $s' = s \cdot \lceil \frac{6}{sI_i} \rceil$. Imagine that we have s' new samples, $U_{i,1}, V_{i,1}, ..., U_{i,s'}, V_{i,s'}$, where each $(U_{i,j}, V_{i,j})$ $(j = 1, \ldots, s')$ is generated independently according to the same distribution as $(X_{i,1}, Y_{i,1})$. Since $s' \geq 6/I_i$, by Corollary 7.6.1, we have that

$$\Pr\left[\text{sign}\left(\sum_{j=1}^{s'}(U_{i,j}-V_{i,j})\right)=(-1)^B\right]>5/6.$$

Write $(U_iV_i)^{[a,b]}$ as shorthand for the sequence $((U_{i,a},V_{i,a}),\ldots(U_{i,b},V_{i,b}))$, and define $(X_iY_i)^{[a,b]}$ analogously. By Fano's inequality, we have that

$$\begin{aligned} I\left((U_iV_i)^{[1,s']};B\right) &= H(B)-H(B|(U_iV_i)^{[1,s']}) \\ &\geq H(\tfrac{1}{2})-H(1-\tfrac{5}{6})=1-H(\tfrac{1}{6})\geq 1/3. \end{aligned}$$

Since $I((U_iV_i)^{[1,s]};(U_iV_i)^{[s+1,s']}|B)=0$ (our new samples are independent given B), we have

$$\begin{aligned} I((U_iV_i)^{[1,s']};B) &= I((U_iV_i)^{[1,s]};B|(U_iV_i)^{[s+1,s']})+I((U_iV_i)^{[s+1,s']};B) \\ &\leq I((U_iV_i)^{[1,s]};B)+I((U_iV_i)^{[s+1,s']};B) \qquad \text{(by Fact 1.2.3)} \end{aligned}$$

Repeating this procedure, we get

$$I((U_iV_i)^{[1,s']};B)\leq\sum_{u=1}^{\lceil\frac{6}{sI_i}\rceil}I((U_iV_i)^{[(u-1)s+1,us]};B).$$

Since we know that for any $u=1,...,\lceil\frac{6}{sI_i}\rceil$,

$$I((U_iV_i)^{[(u-1)s+1,us]};B)=I((X_iY_i)^{[1,s]};B),$$

we get

$$I((X_iY_i)^{[1,s]};B)\geq I((U_iV_i)^{[1,s']};B)\cdot\frac{1}{\lceil\frac{6}{sI_i}\rceil}\geq\frac{sI_i}{6\cdot 6}.$$

The last inequality is true because $\frac{6}{sI_i}=\frac{\sum_{k=1}^{n}I_k}{24\sqrt{n}I_i}\geq 1$.

After we lower bound $I((X_iY_i)^{[1,s]}; B)$, we are going to show that we can output B correctly with reasonable probability based on samples only from coordinate i.

$$\begin{aligned}
\frac{sI_i}{6\cdot 6} &\leq I((X_iY_i)^{[1,s]}; B) \\
&= \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot D(B|(X_iY_i)^{[1,s]} = x \| B) \\
&\leq \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot \big(2(\Pr[B = 0|(X_iY_i)^{[1,s]} = x] - 1/2)^2 + \\
&\quad 2(\Pr[B = 1|(X_iY_i)^{[1,s]} = x] - 1/2)^2\big) \qquad \text{(by Fact 1.2.6)} \\
&= \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot (\Pr[B = 0|(X_iY_i)^{[1,s]} = x] - \Pr[B = 1|(X_iY_i)^{[1,s]} = x])^2 \\
&\leq \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot \big|\Pr[B = 0|(X_iY_i)^{[1,s]} = x] - \Pr[B = 1|(X_iY_i)^{[1,s]} = x]\big| .
\end{aligned}$$

When $\sum_{j=1}^{s}(X_{i,j} - Y_{i,j}) > 0$, it is easy to check that

$$\Pr[B = 0|(X_iY_i)^{[1,s]}] > \Pr[B = 1|(X_iY_i)^{[1,s]}].$$

Therefore,

$$\begin{aligned}
\Pr[S_{i,1} = (-1)^B] &= \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot \\
&\quad \max\big(\Pr[B = 0|(X_iY_i)^{[1,s]} = x], \Pr[B = 1|(X_iY_i)^{[1,s]} = x]\big) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \sum_x \Pr[(X_iY_i)^{[1,s]} = x] \cdot \\
&\quad \big|\Pr[B = 0|(X_iY_i)^{[1,s]} = x] - \Pr[B = 1|(X_iY_i)^{[1,s]} = x]\big| \\
&\geq \frac{1}{2} + \frac{sI_i}{12 \cdot 6} \\
&\geq \frac{1}{2} + \frac{\sqrt{n}I_i}{\sum_{k=1}^{n} I_k}.
\end{aligned}$$

Similarly, we can show for all $i = 1, ..., n$, $j = 1, ..., l$,

$$\Pr[S_{i,j} = (-1)^B] \geq \frac{1}{2} + \frac{\sqrt{n} I_i}{\sum_{k=1}^n I_k}.$$

Now without loss of generality assume that $B = 0$. We have that

$$\mathrm{E}\left[\sum_{i=1}^n S_{i,j}\right] \geq \sum_{i=1}^n \left(\frac{1}{2} + \frac{\sqrt{n} I_i}{\sum_{k=1}^n I_k} - \frac{1}{2} + \frac{\sqrt{n} I_i}{\sum_{k=1}^n I_k}\right) = 2\sqrt{n}.$$

Therefore, by the Chernoff bound,

$$\Pr[T_j = 1] \geq 1 - e^{-(1/n)\cdot(2\sqrt{n})^2\cdot(1/2)} > 3/4.$$

By the Chernoff bound again,

$$\Pr[Z_2 > 0] \geq 1 - e^{-\ell\cdot(1/2)^2\cdot(1/2)} \geq 1 - \frac{\alpha}{2n}.$$

Since we initially fail with probability at most $\frac{\alpha}{2}$, by the union bound, in Case 2 we fail with probability at most $\frac{\alpha}{2} + \frac{\alpha}{2n} < \alpha$. This concludes the proof.

□

7.6.2 $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ with unbounded competitive ratios even for constant n

In this section, we show that the competitive ratios of $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ are unbounded even when n is a constant. In other words, we cannot upper bound the competitive ratios of $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ by only a function of n. The competitive ratio also needs to depend on some parameters of the instance. We prove this by showing instances where the competitive ratios of $\mathcal{A}_{count}$ and $\mathcal{A}_{max}$ also depend on ε which is some parameter of the instances in Lemma 7.6.4 and Lemma 7.6.5. The result in Lemma 7.6.4 can be easily generalized to show that the counting algorithm of [177] for Top-k also has unbounded competitive ratio even when n is a constant. Proofs can be found in the full version of [57].

Lemma 7.6.4. *For each sufficiently large n and for any $\varepsilon > 0$, there exists an instance $C = (n, \mathbf{p}, \mathbf{q})$ of* Domination *such that the following two statements are true:*

1. $r_{min}(C, \mathcal{A}_{coup}, 1 - \frac{2}{n}) \leq \frac{5184\sqrt{n}\log n}{\varepsilon}$

2. $r_{min}(C, \mathcal{A}_{count}) \geq \frac{n}{16\varepsilon^2}$.

Lemma 7.6.5. *For each sufficiently large n and any $0 < \varepsilon < 1/n^3$, there exists an instance $C = (n, \mathbf{p}, \mathbf{q})$ of* Domination *such that the following two statements are true.*

1. $r_{min}(C, \mathcal{A}_{coup}, 1 - \frac{2}{n}) \leq \frac{518400\sqrt{n}\ln n}{\varepsilon}$.

2. $r_{min}(C, \mathcal{A}_{max}, \frac{9}{10}) \geq \frac{1}{\varepsilon^2 2^{14}\ln n}$

7.7 Reducing Top-k to Domination

In this section, we will finally reduce Top-k to Domination, thus proving Theorem 7.3.1. First, we will give an algorithm for Top-k problem that uses $\mathcal{A}_{coup}$ for Domination as a subroutine. We need Lemma 7.7.1 and Lemma 7.7.2 for the algorithm. Their proof can be found in the full version of [57]. We begin by reducing Top-k to the following graph theoretic problem.

Lemma 7.7.1. *Let $G = ([n], E)$ be a directed complete graph on vertices $\{1, 2, \cdots, n\}$ i.e. for every distinct $i, j \in [n]$, either $(i, j) \in E$ or $(j, i) \in E$ but not both. Suppose there is a subset $S \subset [n]$ of size k such that $(i, j) \in E$ for every $i \in S$ and $j \notin S$. Then there is a randomized algorithm which runs in expected running time $O(n)$ and finds the set S given oracle access to the edges of G. Moreover there is some absolute constant $C > 0$ such that for every $\lambda \geq 1$, the probability that the algorithm runs in more than $C\lambda n$ time is bounded by* $\exp(-\lambda)$.

The following lemma shows that when $p \geq q$, $\mathbb{I}(p,q)$ is an increasing function of p and a decreasing function of q.

Lemma 7.7.2. *Let $0 \leq q' \leq q \leq p \leq p' \leq 1$, then $\mathbb{I}(p',q') \geq \mathbb{I}(p,q)$.*

We are now ready to give an algorithm for Top-k.

Theorem 7.7.1. *There exists an algorithm A for* Top-k *such that for any $\alpha > 0$ and any instance $S = (n, k, \mathbf{P})$, A runs in time $O(n^2 r \log(1/\alpha))$ and satisfies*

$$r_{min}(S, A, 1-\alpha) \leq \frac{7776\sqrt{n}\log(2n\alpha^{-1})}{\mathbb{I}(\mathbf{P}_k, \mathbf{P}_{k+1})}$$

where $\mathbf{P}_k, \mathbf{P}_{k+1}$ are the k and $k+1$ rows of $\mathbf{P}$.

Proof. Let $\mathbf{P}_i$ denote the i^{th} row of $\mathbf{P}$, and let $\Delta = I(\mathbf{P}_k, \mathbf{P}_{k+1})$. Recall that $\mathcal{A}$ is given as input the three-dimensional array of samples $Z_{i,j,l}$, where for each $i, j \in [n]$ and $1 \leq l \leq r$, $Z_{i,j,l}$ is the result of the lth noisy comparison between item i and item j (sampled from $\mathcal{B}(\mathbf{P}_{\pi^{-1}(i),\pi^{-1}(j)})$). We will define a complete directed graph $G = ([n], E)$ as follows. For every $1 \leq i < j \leq n$ and $1 \leq h \leq n$, run $\mathcal{A}_{coup}$ with input $X_{h,l} = Z_{i,h,l}$ and $Y_{h,l} = Z_{j,h,l}$; if $\mathcal{A}_{coup}$ returns $B = 0$, then direct the edge from i towards j, and otherwise, direct the edge from j towards i.

Let $T = \{\pi(1), \pi(2), \ldots, \pi(k)\}$ be the set of labels of the top k items. We claim that if $i \in T$ and $j \notin T$, then with probability at least $1 - \frac{\alpha}{n^2}$, the edge is directed from i towards j. To see this, note that in the corresponding input to $\mathcal{A}_{coup}$, X is drawn from $\mathbf{P}_{\pi^{-1}(i)}$ and Y is drawn from $\mathbf{P}_{\pi^{-1}(j)}$. If $i \in T$ and $j \notin T$, then $\pi^{-1}(i) \leq k < \pi^{-1}(j)$. In particular, $\mathbf{P}_{\pi^{-1}(i)}$ dominates $\mathbf{P}_{\pi^{-1}(j)}$, and moreover by Lemma 7.7.2, $\mathbb{I}(\mathbf{P}_{\pi^{-1}(i)}, \mathbf{P}_{\pi^{-1}(j)}) \geq \Delta$. It follows from Theorem 7.6.1 that A_{coup} outputs $B = 0$ on this input with probability at least $1 - \frac{\alpha}{2n^2}$, since in general,

$$
\begin{aligned}
r_{min}(C, \mathcal{A}_{coup}, 1 - \tfrac{\alpha}{2n^2}) &\leq \frac{2592\sqrt{n}\log(4n^3\alpha^{-1})}{\mathbb{I}(\mathbf{p}, \mathbf{q})} \\
&\leq \frac{7776\sqrt{n}\log(2n\alpha^{-1})}{\mathbb{I}(\mathbf{p}, \mathbf{q})}.
\end{aligned}
$$

By the union bound, the probability that all of these comparisons are correct is at least $1-\frac{\alpha}{2}$. Therefore, by the tail bounds in Lemma 7.7.1, we can find the subset T in $O(n\log(1/\alpha))$ oracle calls to $\mathcal{A}_{coup}$ with probability at least $1 - \frac{\alpha}{2}$. The probability of failure is at most $\frac{\alpha}{2} + \frac{\alpha}{2} = \alpha$. Each call to $\mathcal{A}_{coup}$ takes $O(nr)$ time, so the overall time of the algorithm is $O(n^2 r\log(1/\alpha))$.

□

To prove that this algorithm is competitive, we will conclude by proving a lower bound on $r_{min}(S)$ (again, by reduction to the appropriate lower bound for DOMINATION).

Lemma 7.7.3. *Let $S = (n, k, \mathbf{P})$ be an instance of* TOP-k. *Then $r_{min}(S) \geq \frac{0.1}{\mathbb{I}(\mathbf{P}_k, \mathbf{P}_{k+1})}$.*

Proof. We will proceed by contradiction. Suppose there exists an algorithm A which satisfies $r_{min}(S, A) \leq \frac{0.01}{\mathbb{I}(P_k, P_{k+1})}$. We will show how to convert this into an algorithm A' which solves the instance $C = (n, \mathbf{P}_k, \mathbf{P}_{k+1})$ of DOMINATION with probability at least $\frac{3}{4}$ when given at least $2r = 0.05/\mathbb{I}(\mathbf{P}_k, \mathbf{P}_{k+1})$ samples, thus contradicting Lemma 7.4.1.

The algorithm A' is described in Algorithm 11; essentially, A' embeds the inputs X and Y to the DOMINATION instance as rows/columns k and $k+1$ respectively of the TOP-k instance. It is easy to check that the $Z_{i,j,l}$ for $i, j \in [n]$,$l \in [r]$ generated in A' are distributed according to the same distribution as the corresponding elements in the instance S of TOP-k. Therefore A will output the top k items correctly with probability at least $3/4$. In addition, if $B = 0$ the item labeled k will be in the top k items and if $B = 1$ the item labeled k will not be in the top k items. Therefore, A' succeeds to solve this instance of DOMINATION with probability at least $3/4$, leading to our desired contradiction.

Algorithm 11 Algorithm A' for the lower bound reduction

Get input $X_{i,l}$, $Y_{i,l}$ for $i \in [n]$ and $l \in [2r]$ from DOMINATION$(n, \mathbf{P}_k, \mathbf{P}_{k+1}, 2r)$.
Generate a random permutation π on n elements s.t. $\pi(\{k, k+1\}) = \{k, k+1\}$.
for $i \in [n]$, $j \in [n]$, $l \in [r]$ **do**
    If $i = k$, set $Z_{i,j,l} = X_{j,l}$.
    If $i = k+1$, set $Z_{i,j,l} = Y_{j,l}$.
    If $i \notin \{k, k+1\}, j = k$, set $Z_{i,j,l} = X_{i,l+r}$.
    If $i \notin \{k, k+1\}, j = k+1$, set $Z_{i,j,l} = Y_{i,l+r}$.
    If $i \notin \{k, k+1\}, j \notin \{k, k+1\}$, sample $Z_{i,j,l}$ from $\mathcal{B}(\mathbf{P}_{\pi^{-1}(i),\pi^{-1}(j)})$.
end for
Run A on samples $Z_{i,j,l}$, $i, j \in [n]$, $l \in [r]$.
If A said k is amongst the top k items, output $B = 0$. Otherwise output $B = 1$.

□

We are now ready to prove our main upper bound result.

Corollary 7.7.1. *There is an algorithm A for* TOP-k *such that A runs in time $O(n^2 r)$ and on every instance S of* TOP-k *on n items,*

$$r_{min}(S, A) \leq O(\sqrt{n} \log n) r_{min}(S).$$

Proof. Let $S =$TOP-$k(n, k, \mathbf{P}, \cdot)$ be an instance of TOP-k. By Lemma 7.7.3,

$$r_{\min}(S) \geq \frac{0.1}{\mathbb{I}(\mathbf{P}_k, \mathbf{P}_{k+1})}.$$

If A is the algorithm in Theorem 7.7.1 with $\alpha = \frac{1}{4}$ then A runs in time $O(n^2 r)$ and

$$r_{\min}(S, A) \leq O\left(\frac{\sqrt{n} \log n}{\mathbb{I}(\mathbf{P}_k, \mathbf{P}_{k+1})}\right).$$

Combining these two inequalities, we obtain our result.

□

7.8 Lower Bounds for Domination and Top-k

In the previous section we demonstrated an algorithm that solves Top-k on any distribution using at most $\tilde{O}(\sqrt{n})$ times more samples than the optimal algorithm for that distribution (see Corollary 7.7.1). In this section, we show this is tight up to logarithmic factors; for any algorithm, there exists some distribution where that algorithm requires $\tilde{\Omega}(\sqrt{n})$ times more samples than the optimal algorithm for that distribution. Specifically, we show the following lower bound.

Theorem 7.8.1. *For any algorithm A, there exists an instance S of* Top-k *of size n such that $r_{min}(S, A) \geq \Omega\left(\frac{\sqrt{n}}{\log n}\right) r_{min}(S)$.*

As in the previous sections, instead of proving this lower bound directly, we will first prove a lower bound for the domination problem, which we will then embed in a Top-k instance.

Theorem 7.8.2. *For any algorithm A, there exists an instance C of* Domination *of size n such that $r_{min}(C, A) \geq \Omega\left(\frac{\sqrt{n}}{\log n}\right) r_{min}(C)$.*

7.8.1 A hard distribution for domination

To prove Theorem 7.8.2, we will show that there exists a distribution over instances of the domination problem such that, while each instance in the support of this distribution can be solved by some algorithm with a small number of samples, any algorithm requires a large number of samples given an instance randomly sampled from this distribution.

Let $\mathcal{C}$ be a distribution over instances C of the domination problem of size n. We extend r_{min} to distributions by defining $r_{min}(\mathcal{C}, A, p)$ as the minimum number of samples algorithm A needs to successfully solve Domination with probability at least p over instances randomly sampled from $\mathcal{C}$, and let $r_{min}(\mathcal{C}, A) = r_{min}(\mathcal{C}, A, 3/4)$. The following lemma relates the distributional sample complexity to the single instance sample complexity.

Lemma 7.8.1. *For any $p > 1/2$, algorithm A and any distribution $\mathcal{C}$ over instances of the domination problem, there exists a C in the support of $\mathcal{C}$ such that $r_{min}(C, A, p) \geq r_{min}(\mathcal{C}, A, p)$.*

Proof. Let $\varepsilon(C, A, r)$ be the probability that algorithm A errs given r samples from C. By the definition of $r_{\min}(\mathcal{C}, A, p)$, we have that

$$\sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}}[C] \cdot \varepsilon(C, A, r_{min}(\mathcal{C}, A, p)) = 1 - p$$

It follows that there exists some $C^* \in \mathsf{supp}\mathcal{C}$ such that

$$\varepsilon(C^*, A, r_{min}(\mathcal{C}, A, p)) \geq 1 - p$$

Since $\varepsilon(C^*, A, r)$ is decreasing in r, this implies that $r_{min}(C^*, A, p) \geq r_{min}(\mathcal{C}, A, p)$, as desired. $\square$

We will find it useful to work with distributions that are only mostly supported on easy instances. The following lemma lets us do that.

Lemma 7.8.2. *Let $\mathcal{C}$ be a distribution over instances of the domination problem, and let E be an event with $\Pr[E] = 1 - \delta$. Then for any algorithm A and any $1 - \delta > p > \frac{1}{2}$, $r_{min}(\mathcal{C}|E, A, p+\delta) \geq r_{min}(\mathcal{C}, A, p)$ (here $\mathcal{C}|E$ denotes the distribution $\mathcal{C}$ conditioned on event E occurring).*

Proof. By the definition of $r_{\min}(\mathcal{C}, A, p)$, we have that

$$\sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}}[C] \cdot \varepsilon(C, A, r_{min}(\mathcal{C}, A, p)) = 1 - p$$

Rewrite this as

$$\Pr[\overline{E}] \cdot \sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}|\overline{E}}[C] \cdot \varepsilon(C, A, r_{min}(\mathcal{C}, A, p)) + \Pr[E] \cdot \sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}|E}[C] \cdot \varepsilon(C, A, r_{min}(\mathcal{C}, A, p)) = 1 - p$$

Since $\sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}|\overline{E}}[C] = 1$ and $\Pr[\overline{E}] = \delta$, it follows that

$$\sum_{C \in \mathsf{supp}\mathcal{C}} \Pr_{\mathcal{C}|E}[C] \cdot \varepsilon(C, A, r_{min}(\mathcal{C}, A, p)) \geq 1 - p - \delta$$

from which it follows that $r_{min}(\mathcal{C}|E, A, p + \delta) \geq r_{min}(\mathcal{C}, A, p)$.

□

We can now define the hard distribution for the domination problem. Define $\gamma = \frac{1}{100\sqrt{n}}$. Let S_P be a random subset of $[n]$ where each $i \in [n]$ is independently chosen to belong to S_P with probability γ. Likewise, define S_Q the same way (independently of S_P). Finally, fix n constants R_i all in the range $[\frac{1}{4}, \frac{3}{4}]$ (for now, it is okay to consider only the case where $R_i = \frac{1}{2}$ for all i; to extend this lower bound to the top-k problem, we will need to choose different values of R_i). Then the hard distribution $\mathcal{C}_{hard}$ is the distribution over instances $C(S_P, S_Q) = (n, \mathbf{p}, \mathbf{q})$ of DOMINATION where

$$p_i = \begin{cases} R_i(1+\varepsilon) & \text{if } i \in S_P \\ R_i & \text{if } i \notin S_P \end{cases}$$

and

$$q_i = \begin{cases} R_i(1-\varepsilon) & \text{if } i \in S_Q \\ R_i & \text{if } i \notin S_Q \end{cases}$$

We claim that the majority of the instances in the support of $\mathcal{C}_{hard}$ have an algorithm that requires few samples. Intuitively, if S_P and S_Q are fixed, then the best algorithm for that specific instance can restrict attention only to the indices in S_P and S_Q. In particular, if S_P is large enough (some constant times its expected size), then simply throwing away all indices not in S_P and counting which row has more heads is an efficient algorithm for recovering the dominant set.

Theorem 7.8.3. *Fix any S_P and S_Q such that $|S_P| \geq \frac{1}{10}n\gamma$. Then $r_{min}(C(S_P, S_Q), p) = O\left(\frac{\log(1-p)^{-1}}{\varepsilon^2\sqrt{n}}\right)$ for all $p < 1$.*

Proof. It suffices to demonstrate an algorithm A such that $r_{min}(C(S_P, S_Q), A, p) = O\left(\frac{\log(1-p)^{-1}}{\varepsilon^2\sqrt{n}}\right)$.

Any algorithm A receives two sets X, Y, each of r samples from n coins. Write $X = (X_1, X_2, \ldots, X_n)$, where each $X_i = (X_{i,1}, X_{i,2}, \ldots X_{i,r})$ is the collection of r samples from coin i (likewise, write $Y = (Y_1, Y_2, \ldots, Y_n)$, and $Y_i = (Y_{i,1}, Y_{i,2}, \ldots Y_{i,r})$). Consider the following algorithm: A computes the value

$$T = \sum_{i \in S_P} \sum_{j=1}^{r} (X_{i,j} - Y_{i,j})$$

and outputs that $B = 0$ if $T \geq 0$ and outputs $B = 1$ otherwise.

For each i, j, let $A_{i,j} = X_{i,j} - Y_{i,j}$. If $B = 0$, then $A_{i,j} \in [-1, 1]$, $\mathrm{E}[A_{i,j}] \geq \varepsilon R_i \geq \frac{\varepsilon}{4}$ and all the $A_{i,j}$ are independent. It follows from Hoeffding's inequality that in this case,

$$\begin{aligned}
\Pr[T < 0] &= \Pr[T - \mathrm{E}[T] < -\mathrm{E}[T]] \\
&\leq \exp\left(-\frac{2E[T]^2}{4|S_P|r}\right) \\
&= \exp\left(-\frac{|S_P|r\varepsilon^2}{32}\right) \\
&\leq \exp\left(-\frac{\gamma n\varepsilon^2 r}{320}\right) \\
&= \exp\left(-\frac{\sqrt{n}\varepsilon^2 r}{32000}\right)
\end{aligned}$$

Therefore, choosing $r = \frac{32000\ln(1-p)^{-1}}{\sqrt{n}\varepsilon^2} = O\left(\frac{\log(1-p)^{-1}}{\sqrt{n}\varepsilon^2}\right)$ guarantees $\Pr[T < 0] \leq 1 - p$. Similarly, the probability that $T \geq 0$ if $B = 1$ is also at most $1 - p$ for this r. The conclusion follows. □

By a simple Chernoff bound, we also know that the event that S_P has size at least $\frac{1}{10}n\gamma$ occurs with high probability.

Lemma 7.8.3. $\Pr\left[|S_P| \geq \frac{1}{10}n\gamma\right] \geq 1 - e^{-\sqrt{n}/400}$.

In the following subsection, we will prove that for all A, $r_{min}(\mathcal{C}_{hard}, A)$ is large. More precisely, we will prove the following theorem.

Theorem 7.8.4. *For all algorithms A, $r_{min}(\mathcal{C}_{hard}, A, \frac{2}{3}) = \Omega\left(\frac{1}{\varepsilon^2 \log n}\right)$.*

Given that this theorem is true, we can complete the proof of Theorem 7.8.2.

Proof of Theorem 7.8.2. By Theorem 7.8.4, for any algorithm A, $r_{min}(\mathcal{C}_{hard}, A, \frac{2}{3}) = \Omega\left(\frac{1}{\varepsilon^2 \log n}\right)$. Let E be the event that $|S_P| \geq \frac{1}{10}n\gamma$. By Lemma 7.8.3, if $n \geq (400 \ln \frac{12}{11})^2$, $\Pr[E] \geq \frac{1}{12}$. It then follows from Lemma 7.8.2 that

$$\begin{aligned} r_{min}(\mathcal{C}_{hard}|E, A) &= r_{min}(\mathcal{C}_{hard}|E, A, 3/4) \\ &\geq r_{min}(\mathcal{C}_{hard}, A, 2/3) \\ &\geq \Omega\left(\frac{1}{\varepsilon^2 \log n}\right). \end{aligned}$$

It then follows by Lemma 7.8.1 that there is a specific instance $C = C(S_P, S_Q)$ with $|S_P|$ at least $\frac{1}{10}\gamma n$ such that $r_{min}(C, A) \geq \Omega\left(\frac{1}{\varepsilon^2 \log n}\right)$. On the other hand, by Theorem 7.8.3, for this C, $r_{min}(C) \leq O\left(\frac{1}{\varepsilon^2 \sqrt{n}}\right)$. It follows that for any algorithm A, there exists an instance C such that $r_{min}(C, A) \geq \Omega\left(\frac{\sqrt{n}}{\log n}\right) r_{min}(C)$, as desired.

□

7.8.2 Proof of lower bounds

In this subsection, we prove Theorem 7.8.4; namely, we will show that any algorithm needs at least $\Omega\left(\frac{1}{\varepsilon^2 \log n}\right)$ samples to succeed on $\mathcal{C}_{hard}$ with constant probability. Our main approach will be to bound the mutual information between the samples provided to the algorithm and the correct output (recall that B is the hidden bit that determines whether the samples in X are drawn from $\mathbf{p}$ or from $\mathbf{q}$).

Lemma 7.8.4. *If $I(XY;B) < 0.05$, then there is no algorithm that can succeed at identifying B with probability at least $\frac{2}{3}$.*

Proof. Fix an algorithm A, and let p_e be the probability that it errs at computing B. By Fano's inequality, we have that

$$\begin{aligned} H(p_e) &\geq H(B|XY) \\ &= H(B) - I(XY;B) \\ &= 1 - I(XY;B) \\ &> 0.95 \end{aligned}$$

Since $H(\frac{1}{3}) \leq 0.95$, it follows that A must err with probability at least 1/3. □

Via the chain rule, we can decompose $I(XY;B)$ into the sum of many smaller mutual informations.

Lemma 7.8.5. $I(XY;B) \leq \sum_{i=1}^{n} (I(X_i;B) + I(Y_i;B))$

Proof. Write $X^{<i}$ to represent the concatenation $X_1 X_2 \ldots X_{i-1}$. By the chain rule, we have that

$$I(XY;B) = \sum_{i=1}^{n} I(X_i Y_i; B | X^{<i} Y^{<i})$$

We claim that $I(X_i Y_i; X^{<i} Y^{<i} | B) = 0$. To see this, note that given B, each coin in X_i is sampled from some $\mathcal{B}(p)$ distribution, where p only depends on whether $i \in S_P$ or $i \in S_Q$. Since each i is chosen to belong to S_P and S_Q independently with probability γ, this implies

X_i (and similarly Y_i) are independent from $X^{<i}$ and $Y^{<i}$ given B. By Fact 1.2.3, this implies that $I(X_iY_i;B|X^{<i}Y^{<i}) \leq I(X_iY_i;B)$, and therefore that

$$I(XY;B) \leq \sum_{i=1}^{n} I(X_iY_i;B).$$

Likewise, we can write $I(X_iY_i;B) = I(X_i;B) + I(Y_i;B|X_i)$. Since $I(X_i;Y_i|B) = 0$ (since S_P and S_Q are chosen independently), again by Fact 1.2.3 it follows that $I(Y_i;B|X_i) \leq I(Y_i;B)$ and therefore that

$$I(XY;B) \leq \sum_{i=1}^{n} \left(I(X_i;B) + I(Y_i;B)\right).$$

□

Lemma 7.8.6. *If $n \geq 400$ and $r = \frac{1}{100\varepsilon^2 \ln n}$, then for all i, $I(B;X_i) = I(B;Y_i) \leq \frac{1}{100n}$.*

Proof. By symmetry, $I(B;X_i) = I(B;Y_i)$. We will show that $I(B;X_i) \leq \frac{1}{100n}$.

Let $Z_i = \sum_j X_{i,j}$. Note that Z_i is a sufficient statistic for B, and therefore $I(B;X_i) = I(B;Z_i)$. By Fact 1.2.5,

$$\begin{aligned} I(B;Z_i) &= \mathrm{E}_{Z_i}[D(B|Z_i \| B)] \\ &= \sum_{z=0}^{r} \Pr[Z_i = z] \cdot D(\Pr[B=0|Z_i=z] \| \tfrac{1}{2}). \end{aligned}$$

We next divide the range of z into two cases.

1. **Case 1**: $|z - rR_i| \leq 11r\varepsilon \ln n$.

   In this case, we will bound the size of $D(\Pr[B=0|Z_i=z]\|\frac{1}{2})$. Note that

$$
\begin{aligned}
\left|\Pr[B=0|Z_i=z]-\frac{1}{2}\right| &= \left|\frac{\Pr[Z_i=z|B=0]\cdot\Pr[B=0]}{\Pr[Z_i=z]}-\frac{1}{2}\right| \\
&= \left|\frac{\Pr[Z_i=z|B=0]}{\Pr[Z_i=z|B=0]+\Pr[Z_i=z|B=1]}-\frac{1}{2}\right| \\
&= \frac{|\Pr[Z_i=z|B=0]-\Pr[Z_i=z|B=1]|}{2(\Pr[Z_i=z|B=0]+\Pr[Z_i=z|B=1])} \qquad (7.4)
\end{aligned}
$$

Now, note that

$$
\begin{aligned}
\Pr[Z_i=z|B=0] &= (1-\gamma)\binom{r}{z}R_i^z(1-R_i)^{r-z}+\gamma\binom{r}{z}(R_i(1+\varepsilon))^z(1-R_i(1+\varepsilon))^{r-z} \\
\Pr[Z_i=z|B=1] &= (1-\gamma)\binom{r}{z}R_i^z(1-R_i)^{r-z}+\gamma\binom{r}{z}(R_i(1-\varepsilon))^z(1-R_i(1-\varepsilon))^{r-z}
\end{aligned}
$$

We can therefore lower bound the denominator of (7.4) via

$$
\begin{aligned}
2(\Pr[Z_i=z|B=0]+\Pr[Z_i=z|B=1]) &\geq 4(1-\gamma)\binom{r}{z}R_i^z(1-R_i)^{r-z} \\
&\geq 2\binom{r}{z}R_i^z(1-R_i)^{r-z}
\end{aligned}
$$

Likewise, we can write the numerator of (7.4) as

$$
|\Pr[Z_i=z|B=0]-\Pr[Z_i=z|B=1]| = \gamma\binom{r}{z}R_i^z(1-R_i)^{r-z}M
$$

where

$$\begin{aligned} M &= \left|(1+\varepsilon)^z\left(\frac{1-R_i(1+\varepsilon)}{1-R_i}\right)^{r-z} - (1-\varepsilon)^z\left(\frac{1-R_i(1-\varepsilon)}{1-R_i}\right)^{r-z}\right| \\ &= \left|(1+\varepsilon)^z\left(1-\frac{R_i}{1-R_i}\varepsilon\right)^{r-z} - (1-\varepsilon)^z\left(1+\frac{R_i}{1-R_i}\varepsilon\right)^{r-z}\right|. \end{aligned}$$

To bound M, note that (applying the inequality $1+x \leq e^x$)

$$\begin{aligned} (1+\varepsilon)^z\left(1-\frac{R_i}{1-R_i}\varepsilon\right)^{r-z} &\leq \exp\left(\varepsilon z - \varepsilon\frac{R_i}{1-R_i}(r-z)\right) \\ &= \exp\left(\varepsilon\frac{z-rR_i}{1-R_i}\right) \\ &\leq \exp(4\varepsilon(z-rR_i)) \\ &\leq \exp(44r\varepsilon^2\ln n) \\ &= e^{0.44} \\ &< 2 \end{aligned}$$

Similarly, $(1-\varepsilon)^z\left(1+\frac{R_i}{1-R_i}\varepsilon\right)^{r-z} \leq 2$. It follows that $M \leq 2$, and therefore that

$$\begin{aligned} \left|\Pr[B=0|Z_i=z] - \frac{1}{2}\right| &= \frac{|\Pr[Z_i=z|B=0] - \Pr[Z_i=z|B=1]|}{2(\Pr[Z_i=z|B=0] + \Pr[Z_i=z|B=1])} \\ &\leq \frac{\gamma\binom{r}{z}R_i^z(1-R_i)^{r-z}M}{2\binom{r}{z}R_i^z(1-R_i)^{r-z}} \\ &= \frac{\gamma M}{2} \\ &\leq \gamma \end{aligned}$$

By Fact 1.2.6, this implies that

$$D(\Pr[B=0|Z_i=z]\|\tfrac{1}{2}) \leq \frac{4\gamma^2}{\ln 2}.$$

2. **Case 2**: $|z - rR_i| > 11r\varepsilon \ln n$.

   Let Z^+ be the sum of r i.i.d. $\mathcal{B}(R_i(1+\varepsilon))$ random variables. Note that since Z is the sum of r $\mathcal{B}(p)$ random variables for some $p \leq R_i(1+\varepsilon)$, $\Pr[Z^+ \geq x] \geq \Pr[Z \geq x]$ for all x. Therefore, by Hoeffding's inequality, we have that

$$\begin{aligned}
\Pr\left[Z - rR_i \geq 11r\varepsilon \ln n\right] &\leq \Pr\left[Z^+ - rR_i \geq 11r\varepsilon \ln n\right] \\
&\leq \Pr\left[Z^+ - rR_i(1+\varepsilon) \geq r\varepsilon(11 \ln n - R_i)\right] \\
&\leq \Pr\left[Z^+ - \mathrm{E}[Z^+] \geq 10r\varepsilon \ln n\right] \\
&\leq \exp\left(-\frac{2(10r\varepsilon \ln n)^2}{r}\right) \\
&= \exp(-2\ln n) \\
&= n^{-2}
\end{aligned}$$

   Likewise, we can show that

$$\Pr\left[Z - rR_i \leq -11r\varepsilon \ln n\right] \leq n^{-2}$$

   so

$$\Pr\left[|Z - rR_i| \geq 11r\varepsilon \ln n\right] \leq 2n^{-2}$$

Combining these two cases, we have that (for $n \geq 400$)

$$
\begin{aligned}
I(B; Z_i) &= \sum_{z=0}^{r} \Pr[Z_i = z] \cdot D(\Pr[B = 0 | Z_i = z] \| \tfrac{1}{2}) \\
&\leq \sum_{|\|z\| - r/2| > 11 r\varepsilon \ln n} \Pr[Z_i = z] \cdot 1 + \sum_{|\|z\| - r/2| \leq 11 r\varepsilon \ln n} \Pr[Z_i = z] \cdot O(\gamma^2) \\
&\leq 2n^{-2} + \frac{4\gamma^2}{\ln 2} \\
&\leq \frac{1}{100n}.
\end{aligned}
$$

□

We can now complete the proof of Theorem 7.8.4.

Proof of Theorem 7.8.4. Combining Lemmas 7.8.5 and 7.8.6, we have that if $r = \frac{1}{100\varepsilon^2 \ln n}$, then (for $n \geq 400$) $I(XY; B) \leq 2nI(X_i; B) \leq 0.02$. Therefore by Lemma 7.8.4, there exists no algorithm A that, given this number of samples, correctly identifies B (and thus solves the domination problem) with probability at least 2/3. It follows that

$$
r_{min}(\mathcal{C}_{hard}, A, \tfrac{2}{3}) \geq \frac{1}{100\varepsilon^2 \ln n} = \Omega\left(\frac{1}{\varepsilon^2 \log n}\right)
$$

as desired.

□

7.8.3 Proving lower bounds for Top-K

We will now show how to use our hard distribution of instances of Domination to generate a hard distribution of instances of Top-k. Our goal will be to embed our Domination instance as rows k and $k+1$ of our SST matrix; hence, intuitively, deciding which of the two rows (k or $k+1$) belongs to the top k is as hard as solving the domination problem.

Unfortunately, the SST condition imposes additional structure that prevents us from directly embedding any instance of the domination problem. However, for appropriate choices of the constants R_i, all instances in the support of $\mathcal{C}_{hard}$ give rise to valid SST matrices.

Specifically, we construct the following distribution $\mathcal{S}_{hard}$ over TOP-k instances S of size $n+2$. Consider the distribution $\mathcal{C}_{hard}$ over DOMINATION instances of size n, where for $1 \leq i \leq n$, $R_i = \frac{1}{4} + \frac{i}{8n}$, and $\varepsilon = \frac{1}{100n^2}$. Now, consider the following map f from DOMINATION instances $C = (\mathbf{p}, \mathbf{q})$ to TOP-k instances $S = f(C) = (n+2, k, \mathbf{P})$: we choose $k = n+1$ (so that the problem becomes equivalent to identifying row $n+2$) and define the matrix $\mathbf{P}$ as follows:

$$\mathbf{P}_{ij} = \begin{cases} \mathbf{p}_j & \text{if } i = n+1 \text{ and } j \leq n \\ \mathbf{q}_j & \text{if } i = n+2 \text{ and } j \leq n \\ 1 - \mathbf{p}_i & \text{if } j = n+1 \text{ and } i \leq n \\ 1 - \mathbf{q}_i & \text{if } j = n+2 \text{ and } i \leq n \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

In general, for arbitrary $\mathbf{p}$ and $\mathbf{q}$, this matrix may not be an SST matrix. Note however that for this choice of R_i and ε, it is always the case that $R_i(1+\varepsilon) \leq R_{i+1}(1-\varepsilon)$, so for all i (regardless of sample C), $p_i < p_{i+1}$. In addition, all the R_i belong to $[1/4, 3/8]$, so for all i, p_i and q_i are less than $1/2$. From these two observations, it easily follows that if C belongs to the support of $\mathcal{C}_{hard}$, $\mathbf{P}$ is an SST matrix, and $f(C)$ is a valid instance of the top-k problem. We will write $\mathcal{S}_{hard} = f(\mathcal{C}_{hard})$ to denote the distribution of instances of top-k $f(C)$ where C is sampled from $\mathcal{C}_{hard}$. Likewise, for any event E (e.g. the event that $|S_P| \geq \frac{1}{10}n\gamma$) , we write $\mathcal{S}_{hard}|E$ to denote the distribution $f(\mathcal{C}_{hard}|E)$.

We will begin by showing that, if there exists a sample efficient algorithm for some DOMINATION instance C in the support of $\mathcal{C}_{hard}$, there exists a similarly efficient algorithm for the corresponding TOP-k instance $S = f(C)$.

Lemma 7.8.7. *If $C \in \mathsf{supp}\mathcal{C}_{hard}$ and $S = f(C)$, then $r_{min}(S) \leq \max(r_{min}(C, \frac{4}{5}), 1000n^2(1 + \ln n))$.*

Proof. Let A be an algorithm that successfully solves the DOMINATION instance C with probability at least $\frac{4}{5}$ using $r_{min}(C, \frac{4}{5})$ samples. We will show how to use A to construct an algorithm A' that solves the TOP-k instance S with probability at least 3/4 using $r = \max(r_{min}(C, \frac{4}{5}), 1000n^2(1 + \ln n))$ samples.

For each i, j, write $Z_{i,j} = \sum_{\ell=1}^{r} Z_{i,j,\ell}$. Our algorithm A' operates as follows.

1. We begin by finding the two rows with the smallest row sums $\sum_j Z_{i,j}$. Let these two rows have indices c and d. We claim that, with high probability, $\pi^{-1}(\{c, d\}) = \{n + 1, n + 2\}$.

   To see this, note that for all $i \notin \pi(\{n+1, n+2\})$, $\mathbf{P}_{i,j} \geq \frac{1}{2}$, so $\mathrm{E}\left[\sum_j Z_{i,j}\right] \geq \left(\frac{n}{2} + 1\right) r$. Thus, for any fixed $i \notin \pi(\{n + 1, n + 2\})$, it follows from Hoeffding's inequality that

   $$\Pr\left[\sum_j Z_{i,j} \leq \left(\frac{7}{16}n + 1\right) r\right] \leq \exp\left(-\frac{nr}{128}\right)$$

   so by the union bound, the probability that there exists an $i \notin \pi^{-1}(\{n + 1, n + 2\})$ such that $\sum_j Z_{i,j} \leq \left(\frac{7}{16}n + 1\right) r$ is at most $n \exp\left(-\frac{nr}{128}\right)$.

   On the other hand, if $i \in \pi(\{n+1, n+2\})$ then $\mathbf{P}_{i,j} \leq \frac{3}{8}(1+\varepsilon)$ unless $j \in \pi(\{n+1, n+2\})$, where $\mathbf{P}_{i,j} = \frac{1}{2}$; it follows that in this case, $\mathrm{E}\left[\sum_j Z_{i,j}\right] \leq \left(\frac{3n}{8}(1 + \varepsilon) + 1\right) r$. Similarly, applying Hoeffding's inequality in this case, we find that for any fixed $i \in \pi^{-1}(\{n + 1, n + 2\})$,

   $$\Pr\left[\sum_j Z_{i,j} \geq \left(\frac{7}{16}n + 1\right) r\right] \leq \exp\left(-\frac{nr}{128(1 + \varepsilon)^2}\right) \leq 1.5 \exp\left(-\frac{nr}{128}\right)$$

   and thus the probability that there exists some $i \in \pi^{-1}(\{n + 1, n + 2\})$, such that $\sum_j Z_{i,j} \geq \left(\frac{7}{16}n + 1\right) r$ is at most $3 \exp\left(-\frac{nr}{128}\right)$. It follows that, altogether, the probability that $\pi^{-1}(\{c,d\}) \neq \{n+1, n+2\}$ is at most $(n+3)\exp\left(-\frac{nr}{128}\right)$. Since $r \geq 1000n^2 \ln n$, this is at most $4\exp(-1000/128) < 0.01$.

2. We next sort the values $Z_{c,j}$ for $j \in [n+2] \setminus \{c,d\}$ and obtain indices $j_1, j_2, \ldots, j_n$ so that $Z_{c,j_1} \leq Z_{c,j_2} \leq \cdots \leq Z_{c,j_n}$. We claim that, with high probability, for all a, $\pi^{-1}(j_a) = a$.

   For each i, let U_i be the interval $\left[R_i(1-\varepsilon) - \frac{1}{20n}, R_i(1+\varepsilon) + \frac{1}{20n}\right]$. Note that, by our choice of R_i and ε, all the intervals U_i are disjoint, with U_i less than U_{i+1} for all i. We will show that with high probability, $\frac{1}{r}Z_{c,\pi(i)} \in U_i$ for all i, thus implying the previous claim.

   Note that $Z_{c,\pi(i)}$ is the sum of r $\mathcal{B}(p)$ random variables, where p is either $(1+\varepsilon)R_i$, R_i, or $(1-\varepsilon)R_i$. By Hoeffding's inequality, it follows that

$$\begin{aligned}\Pr\left[Z_{c,\pi(i)} \geq r\left(R_i(1+\varepsilon) + \frac{1}{20n}\right)\right] &\leq \exp\left(-2\frac{(r/20n)^2}{r}\right) \\ &= \exp\left(-\frac{r}{200n^2}\right)\end{aligned}$$

   Likewise,

$$\Pr\left[Z_{c,\pi(i)} \leq r\left(R_i(1-\varepsilon) - \frac{1}{20n}\right)\right] \leq \exp\left(-\frac{r}{200n^2}\right)$$

   Thus, for any fixed i,

$$\Pr\left[\frac{Z_{c,\pi(i)}}{r} \notin U_i\right] \leq 2\exp\left(-\frac{r}{200n^2}\right)$$

   and by the union bound, the probability this fails for some i is at most $2n\exp\left(-\frac{r}{200n^2}\right)$. Since $r \geq 1000n^2(1+\ln n)$, $\exp\left(-\frac{r}{200n^2}\right) \leq (ne)^{-5}$, so this probability is at most $2e^{-5} < 0.02$.

3. Finally, we give algorithm A as input $X_{i,\ell} = Z_{c,j_i,\ell}$ and $Y_{i,\ell} = Z_{d,j_i,\ell}$. Note that (conditioned on the above two claims holding), this input is distributed equivalently to input from the Domination instance C. In particular, if $\pi^{-1}(c) = n+1$ and $\pi^{-1}(d) = n+2$, then each $X_{i,\ell}$ is distributed according to $\mathcal{B}(\mathbf{p}_i)$ and each $Y_{i,\ell}$ is distributed according to $\mathcal{B}(\mathbf{q}_i)$, and if $\pi^{-1}(c) = n+2$ and $\pi^{-1}(d) = n+1$, then each $X_{i,\ell}$ is distributed according to $\mathcal{B}(\mathbf{q}_i)$ and each $Y_{i,\ell}$ is distributed according to $\mathcal{B}(\mathbf{p}_i)$. Thus, if A returns $B = 0$, we return $[n+2] \setminus \{d\}$ as the top $n+1$ indices, and if A returns $B = 1$, we return $[n+2] \setminus \{c\}$ as the top $n+1$ indices.

   The probability that A fails given that steps 1 and 2 succeed is at most 0.2, and the probability that either of the two steps fail to succeed is at most $0.01 + 0.02 = 0.03$. Since $0.2 + 0.03 < \frac{1}{4}$, A' succeeds with probability at least $\frac{3}{4}$, as desired.

□

Corollary 7.8.1. *Let E be the event that $|S_P| \geq \frac{1}{10}n\gamma$. If $C \in \mathsf{supp}(\mathcal{C}_{hard}|E)$ and $S = f(C)$, then $r_{min}(S) \leq O(n^{3.5})$.*

Proof. Recall that by Theorem 7.8.3, for any $C \in \mathsf{supp}(\mathcal{C}_{hard}|E)$, $r_{min}(C, \frac{4}{5}) \leq O\left(\frac{1}{\sqrt{n}\varepsilon^2}\right) = O(n^{3.5})$. By Lemma 7.8.7, $r_{min}(S) \leq \max(r_{min}(C, \frac{4}{5}), 1000n^2(1+\ln n)) \leq O(n^{3.5})$. □

We next show that solving Top-k over the distribution $\mathcal{S}_{hard}|E$ is at least as hard as solving Domination over the distribution $\mathcal{C}_{hard}|E$.

Lemma 7.8.8. *For any algorithm A that solves Top-k, there exists an algorithm A' that solves domination such that $r_{min}(\mathcal{S}_{hard}, A, p) \geq \frac{1}{2}r_{min}(\mathcal{C}_{hard}, A', p)$.*

Proof. We will show more generally that for any distribution $\mathcal{C}$ of Domination instances, if $\mathcal{S} = f(\mathcal{C})$ is a valid distribution of Top-k instances, then $r_{min}(\mathcal{S}, A, p) \geq \frac{1}{2}r_{min}(\mathcal{C}, A', p)$.

We will construct A' by embedding the domination instance inside a top-k instance in much the same way that the function f does, and then using A to solve the top-k instance. We receive as input two sets of samples $X_{i,\ell}$ and $Y_{i,\ell}$ (where $1 \leq i, j \leq n$ and $1 \leq \ell \leq r$) from some DOMINATION instance C drawn from $\mathcal{C}$. We then generate a random permutation π of $[n+2]$. We use our input and this permutation to generate a matrix $Z_{i,j,\ell}$ (where $1 \leq i, j \leq n+2$ and $1 \leq \ell \leq \frac{r}{2}$) of samples to input to A as follows.

For $1 \leq i, j \leq n$, set each $Z_{\pi(i),\pi(j),\ell}$ to be a random $\mathcal{B}(\frac{1}{2})$ random variable. Similarly, for $n+1 \leq i, j \leq n+2$, set each $Z_{\pi(i),\pi(j),\ell}$ to be a random $\mathcal{B}(\frac{1}{2})$ random variable. Now, for all $1 \leq j \leq n$, set $Z_{\pi(n+1),\pi(j),\ell} = X_{j,\ell}$ and set $Z_{\pi(n+2),\pi(j),\ell} = Y_{j,\ell}$. Similarly, for all $1 \leq i \leq n$, set $Z_{\pi(i),\pi(n+1),\ell} = 1 - X_{i,\ell+r/2}$ and set $Z_{\pi(i),\pi(n+2),\ell} = 1 - Y_{i,\ell+r/2}$. Finally, set $k = n+1$ and ask A to solve the TOP-k instance defined by k and $Z_{i,j,\ell}$. If A returns that $\pi(n+1)$ is in the top $n+1$ indices, return $B = 0$, and otherwise return $B = 1$.

From our construction, if the r samples of X and Y are distributed according to a DOMINATION instance C, then the $r/2$ samples of Z are distributed according to the TOP-k instance $S = f(C)$. Since A succeeds with probability p on distribution $\mathcal{S}$ with $r_{min}(\mathcal{S}, A, p)$ samples, A' therefore succeeds with probability p on distribution $\mathcal{C}$ with $2r_{min}(\mathcal{S}, A, p)$ samples, thus implying that $r_{min}(\mathcal{S}, A, p) \geq \frac{1}{2} r_{min}(\mathcal{C}, A', p)$. □

Corollary 7.8.2. *For all algorithms A that solve* TOP-k, $r_{min}(\mathcal{S}_{hard}, A, \frac{2}{3}) = \Omega\left(\frac{n^4}{\log n}\right)$.

Proof. Theorem 7.8.4 tells us that for all algorithms A' that solve DOMINATION, $r_{min}(\mathcal{C}_{hard}, A, \frac{2}{3}) = \Omega\left(\frac{1}{\varepsilon^2 \log n}\right) = \Omega\left(\frac{n^4}{\log n}\right)$. Combining this with Lemma 7.8.8, we obtain the desired result. □

We can now prove Theorem 7.8.1 in much the same fashion as Theorem 7.8.2.

Proof of Theorem 7.8.1. By Corollary 7.8.2, $r_{min}(\mathcal{S}_{hard}, A, \frac{2}{3}) = \Omega\left(\frac{n^4}{\log n}\right)$. Let E be the event that $|S_P| \geq \frac{1}{10} n\gamma$ (in the original DOMINATION instance C). By Lemma 7.8.3, if $n \geq (400 \ln \frac{12}{11})^2$, $\Pr[E] \geq \frac{1}{12}$, and it follows from Lemma 7.8.2 that

$$\begin{aligned} r_{min}(\mathcal{S}_{hard}|E, A) &= r_{min}(\mathcal{S}_{hard}|E, A, \frac{3}{4}) \\ &\geq r_{min}(\mathcal{S}_{hard}, A, \frac{2}{3}) \\ &\geq \Omega\left(\frac{n^4}{\log n}\right) \end{aligned}$$

It therefore follows from 7.8.1 that there is a specific instance S in the support of $\mathcal{S}_{hard}|E$ such that $r_{min}(S, A) \geq \Omega\left(\frac{n^4}{\log n}\right)$. However, by Corollary 7.8.1, $r_{min}(S) \leq O(n^{3.5})$. It follows that for any algorithm A, there exists an instance S of Top-k such that $r_{min}(S, A) \geq \Omega\left(\frac{\sqrt{n}}{\log n}\right) r_{min}(S)$, as desired. □

Chapter 8

Top-k Ranking under the Multinomial Logit Model

The results of this chapter are based on joint work with Xi Chen and Zhiyuan Li [59].

8.1 Introduction

The problem of inferring a ranking over a set of n items (e.g., products, movies, URLs) is an important problem in machine learning and finds numerous applications in recommender systems, web search, social choice, and many other areas. To learn the global ranking, an effective way is to present at most l ($l \geq 2$) items at each time and ask about the most favorable item among the given items. Then, the answers from these multi-wise comparisons will be aggregated to infer the global ranking. When the number of items n becomes large, instead of inferring the global ranking over all the n items, it is of more interest to identify the top-k items with a pre-specified k. In this paper, we study the problem of active top-k ranking from multi-wise comparisons, where the goal is to adaptively choose at most l items for each comparison and accurately infer the top-k items with the minimum number of comparisons (i.e., the minimum sample complexity). As an illustration, let us consider a practical scenario: an online retailer is facing the problem of choosing k best designs of handbags among n candidate designs. One popular way is to display several designs to each arriving customer and observe which handbag is chosen. Since a shopping website has a capacity on the maximum number of display spots, each comparison will involve at most l possible designs.

Given the wide application of top-k ranking, this problem has received a lot of attention in recent years, e.g., [177, 179] (please see Section 6.1.1 for more details). Our work greatly extends the existing literature on top-k ranking in the following three directions:

1. Most existing work studies a non-active ranking aggregation problem, where the answers of comparisons are provided statically or the items for each comparison are chosen completely at random. Instead of considering a passive ranking setup, we propose an active ranking algorithm, which adaptively chooses the items for comparisons based on previously collected information..

2. Most existing work chooses some specific function (call this function f) of problem parameters (e.g., n, k, l and preference scores) and shows that the algorithm's sample complexity is at most f. For the optimality, they also show that for any value of f, there exists an instance whose sample complexity equals to that value and any algorithm needs at least $\Omega(f)$ comparisons on this instance. However, this type of algorithms could perform poorly on some instances other than those instances for establishing lower bounds (see examples from [57]); and the form of function f can vary the designed algorithm a lot.

   To address this issue, we establish a much more refined upper bound on the sample complexity. The derived sample complexity matches the lower bound when all the parameters (including the set of underlying preference scores for items) are given to the algorithm. They together show that our lower bound is tight and also our algorithm is nearly instance optimal (see Definition 8.1.1 for the definition of nearly instance optimal).

3. Existing work mainly focuses on pairwise comparisons. We extend the pairwise comparison to the multi-wise comparison (at most l items) and further quantify the role of l in the sample complexity. From our sample complexity result (see Section 7.3), we show that the pairwise comparison could be as helpful as multi-wise comparison unless the underlying instance is very easy.

8.1.1 Model

In this paper, we adopt the widely used multinomial logit (MNL) model [145, 149, 184] for modeling multi-wise comparisons. In particular, we assume that each item i has an underlying preference score (a.k.a. utility in economics) μ_i for $i = 1, \ldots, n$. These scores, which are unknown to the algorithm, determine the underlying ranking of the items. Specifically, $\mu_i > \mu_j$ means that item i is preferred to item j and item i should have a a higher rank. Without loss of generality, we assume that $\mu_1 \geq \mu_2 \geq \cdots \mu_k > \mu_{k+1} \geq \cdots \geq \mu_n$, and thus the true top-$k$ items are $\{1, \ldots, k\}$. At each time t from 1 to T, the algorithm chooses a subset of items with at least two items, denoted by $S_t \subseteq \{1, ..., n\}$, for query/comparison. The size of the set S_t is upper bounded by a pre-fixed parameter l, i.e., $2 \leq |S_t| \leq l$.

Given the set S_t, the agent will report her most preferred item $a \in S_t$ following the multinomial logit (MNL) model:

$$\Pr[a|S_t] = \frac{\exp(\mu_a)}{\sum_{j \in S_t} \exp(\mu_j)}. \tag{8.1}$$

When the size of S_t is 2 (i.e., $l = 2$), the MNL model reduces to Bradley-Terry model [33], which has been widely studied in rank aggregation literature in machine learning (see, e.g., [159, 125, 169, 60]).

In fact, the MNL model has a simple probabilistic interpretation as follows [184]. Given the set S_t, the agent draws her valuation $\nu_j = \mu_j + \epsilon_j$ for each item $j \in S_t$, where μ_j is the mean utility for item j and each ϵ_j is independently, identically distributed random variable following the Gumbel distribution. Then, the probability that $a \in S_t$ is chosen as the most favorable item is $\Pr(\nu_a \geq \nu_j, \forall j \in S_t \backslash \{a\})$. With some simple algebraic derivation using the density of Gumbel distribution (see Chapter 3.1 in [184]), the choice probability $\Pr(\nu_a \geq \nu_j, \forall j \in S_t \backslash \{a\})$ has an explicit expression in (8.1). For notational convenience, we define $\theta_j = \exp(\mu_j)$ for $i = 1, ..., n$, and the choice probability in (8.1) can be equivalently written as $\Pr[a|S_t] = \frac{\theta_a}{\sum_{j \in S_t} \theta_j}$. By adaptively querying the set S_t for $1 \leq t \leq T$ and observing the reported most favorable item in S_t, the goal is to identify the set of top-k items with high probability using the minimum number of queries.

For notation convenience, we assume the i-th item (with the preference score θ_i) is labeled as $\pi_i \in \{1, \ldots, n\}$ by the algorithm at the beginning. Since the algorithm has no prior knowledge on the ranking of items before it makes any comparison, the ranking of the items should have no correlation with the labels of the items. Therefore, $\pi = (\pi_1, ..., \pi_n)$ is distributed as a uniform permutation of $\{1, ..., n\}$.

The notion of instance optimal was originally defined and emphasized as an important concept in [96]. With the MNL model in place, we provide a formal definition of nearly instance optimal in our problem. To get a definition of instance optimal in our problem, we can just replace $\tilde{O}$ with O in Definition 8.1.1. The "nearly" here just means we allow polylog factors.

Definition 8.1.1 (Nearly Instance Optimal). *Given instance $(n, k, l, \theta_1, ..., \theta_n)$, define $c(n, k, l, \theta_1, ..., \theta_n)$ to be the sample complexity of an optimal adaptive algorithm on the instance. We say that an algorithm A is nearly instance optimal, if for any instance $(n, k, l, \theta_1, ..., \theta_n)$, the algorithm A outputs the top-k items with high probability (with probability $1 - 1/n^c$ for some constant c) and only uses at most $\tilde{O}(c(n, k, l, \theta_1, ..., \theta_n))$ number of comparisons. (Note that $\tilde{O}(\cdot)$ hides polylog factors of n and $\frac{1}{\theta_k - \theta_{k+1}}$.)*

8.1.2 Main results

Under the MNL model described in Section 8.1.1, the main results of this paper include the following upper and lower bounds on the sample complexity.

Theorem 8.1.1. *We design an active ranking algorithm which uses*

$$\tilde{O}\Bigg(\frac{n}{l} + k + \frac{\sum_{i \ge k+1} \theta_i}{\theta_k} + \sum_{i \ge k+1, \theta_i \ge \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \le k, \theta_i \le 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2} \Bigg)$$

comparisons with the set size at most l (can be 2-wise, 3-wise,...,l-wise comparisons) to identify the top-k items with high probability (with probability $1 - 1/n^c$ for some constant c).

We note that in Theorem 8.1.1, the notation $\tilde{O}(\cdot)$ hides polylog factors of n and $\frac{1}{\theta_k - \theta_{k+1}}$.

Next, we present a matching lower bound result, which shows that our sample complexity in Theorem 8.1.1 is nearly instance optimal.

Theorem 8.1.2. *For any (possibly active) ranking algorithm A, suppose that A uses comparisons of set size at most l. Even when the algorithm A is given the values of $\{\theta_1, ..., \theta_n\}$ (note that A does not know which item takes the preference score θ_i for each i), A still needs*

$$\Omega\Bigg(\frac{n}{l} + k + \frac{\sum_{i \ge k+1} \theta_i}{\theta_k} + \sum_{i \ge k+1, \theta_i \ge \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \le k, \theta_i \le 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2} \Bigg)$$

comparisons to identify the top-k items with probability at least $7/8$.

Remark 8.1.1. *Notice that our lower bound is with constant probability which is stronger than a with high probability lower bound and therefore it can be matched with our upper bound (up to poly logarithmic factors for the number of comparisons). Also notice our asymptotic* *notation is on n and we don't make any restrictions on other parameters. For example, Theorem 8.1.2's asymptotic notation can be stated as there exists a constant c, for any n larger than some large enough constant, for any $l, k, \theta_1, ..., \theta_n$, A needs*

$$c \cdot \left(\frac{n}{l} + k + \frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \leq k, \theta_i \leq 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2} \right)$$

comparisons to identify the top-k items with probability at least 7/8*.*

Here we give some intuitive explanations of the terms in the above bounds before introducing the proof overview:

1. Term $\frac{n}{l}$: Since each comparison has size at most l, we need at least $\frac{n}{l}$ comparisons to query each item at least once.

2. Term k: As the proof will suggest, in order to find the top-k items, we need to observe most items in the top-k set as chosen items from comparisons. However, we do not have to observe most items in the bottom-$(n-k)$ set. Therefore, there is no term $n-k$ in the bound.

3. Term $\frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \leq k, \theta_i \leq 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}$: Roughly speaking, when $i > k$ and $\theta_i \geq \theta_k/2$, $\Theta\left(\frac{(\theta_k - \theta_i)^2}{\theta_k^2}\right)$ is the amount of information that the comparison between item i and item k reveals. So intuitively, we need $\Omega\left(\frac{\theta_k^2}{(\theta_k - \theta_i)^2}\right)$ to tell that item i ranks after item k. Other quantity can also be understood from an information theoretic perspective.

It is also worthwhile to note that when l is a constant, it's easy to check that

$$\begin{aligned}
&\frac{n}{l} + k + \frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \\
&\sum_{i: i \leq k, \theta_i \leq 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2} \\
= \quad & O\left(\sum_{i=k+1}^{n} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_i^2}{(\theta_{k+1} - \theta_i)^2}\right).
\end{aligned}$$

This is a simpler expression of the instance optimal sample complexity when l is a constant.

Based on the sample complexity results in Theorem 8.1.1 and 8.1.2, we summarize the main theoretical contribution of this paper:

1. We design an active ranking algorithm for identifying top-k items under the popular MNL model. We further prove a matching lower bound, which establishes that the proposed algorithm is nearly instance optimal.

2. Our result shows that the improvement of the multi-wise comparison over the pairwise comparison depends on the difficulty of the underlying instance. Note that the only term in the sample complexity involving l is $\frac{n}{l}$. Therefore, the multi-wise comparison makes a significant difference from the pairwise comparison only when $\frac{n}{l}$ is the leading term in the sample complexity.

   Therefore, unless the underlying instance is really easy (e.g., the instance-adaptive term $k + \frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \leq k, \theta_i \leq 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}$ is $o(n)$. One implication is that most of the θ_i's among $\theta_{k+1}, ..., \theta_n$ are much smaller than θ_k), the pairwise comparison is as helpful as the multi-wise comparison.

8.1.3 Proof overview

In this section, we give some very high level overviews of how we prove Theorem 8.1.1 and Theorem 8.1.2.

Algorithms

To prove Theorem 8.1.1, we consider two separate cases: $l = O(\log n)$ or $l = \Omega(\log n)$.

1. Our result indicates that the only term involving the size of the comparisons l is $\frac{n}{l}$. Therefore, in the first case, by losing a log-factor, we can just focus on only using pairwise comparisons (because of Claim 8.4.2). Our algorithm (12) first randomly select $\tilde{O}(n)$ pairs and proceed by querying all of them once per iteration. After getting the query results, by a standard binomial concentration bound, we are able to construct a confident interval of $\frac{\theta_i}{\theta_i+\theta_j}$ for each pair (i, j) selected by the algorithm in the beginning. In a high level, our algorithm goes by declaring $\theta_i \geq \theta_j$ for pair i, j, if the lower bound of the corresponding confident interval is bigger or equal to 1, or if there already exists d items $(i = i_1), i_2, \ldots, (i_d = j)$ such that we have already declared $\theta_{i_r} \geq \theta_{i_{r+1}}$ for all $r \in [d-1]$. Moreover, it is a well known result in graph theory that if we select $\Omega(n \log n)$ pairs and create an edge between each of them, then with high probability, the resulting graph is an expander. Thus, we can pick d to be as small as $O(\log n)$. With this, we are able to show that, if $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$, then for all $i, j \in [n]$ with $j \geq i + \frac{n}{4}$, the algorithm will successfully declare $\theta_i \geq \theta_j$ after $O\left(\sum_{i=k+1}^{n} \frac{\theta_k^2}{(\theta_k-\theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_i^2}{(\theta_{k+1}-\theta_i)^2}\right)$ many total queries. Thus, we can remove at least $\frac{n}{4}$ items and recurse on a smaller set.

2. The *more* interesting case is when $l = \Omega(\log n)$. To illustrate the idea, let us just consider top 1 item identification ($k = 1$). We consider two cases: (a). $\theta_1 = 1$ and all the other θ_i's are equal but very close to zero. In this case, if we still use pairwise comparisons, then unless we exactly pick item 1 in one of comparisons, we will get no information about which one is the top item. Thus, we still need at least $\Omega(n)$ queries. On the other hand, if we use comparisons of size n, then, after $\Omega(\log n)$ queries, we should be seeing item 1 all the time, and hence be confident enough to declare item 1 as the top one. (b). When all θ_i's are within a constant factor, then, even if we use larger comparison size, we will still see all the θ_i's show up with com-

parable amount of times, and thus it is not beneficial to use comparisons of size > 2. As illustrate in our bound, it is only beneficial to use multi-wise comparisons when $\frac{\sum_{i\geq 2}\theta_i}{\theta_1} + \sum_{i\geq 2, \theta_i \geq \frac{\theta_1}{2}} \frac{\theta_1^2}{(\theta_1-\theta_i)^2} = o(n)$. This implies that $\frac{\sum_{i\geq 2}\theta_i}{\theta_1} = o(n)$ and therefore among $\theta_2, ..., \theta_n$, there are more than half of θ_i's whose value is smaller than some constant fraction of θ_1. Thus, intuitively, like we mentioned in (a), if we select a random subset of items that contains θ_1 and keep querying this set, then, instead of seeing all items in this set with roughly equal probability, we will be seeing item 1 much more often than the median of frequencies of items in the set. Thus, our algorithm can select an item if it "appears very often when querying a set containing it". We will show that, if the number of total queries is

$$\Omega\left(\frac{n}{l} + \frac{\sum_{i\geq 2}\theta_i}{\theta_1} + \sum_{i\geq 2, \theta_i \geq \frac{\theta_1}{2}} \frac{\theta_1^2}{(\theta_1-\theta_i)^2}\right),$$

then we will be able to select all the top 1-items while not selecting any of the bottom $n/2$ items. Thus, we can remove at least $\frac{n}{2}$ items and recurse on a smaller set.

Lower bounds

To prove Theorem 8.1.2, we establish several lower bounds and combine them using a simple averaging argument. Most of these lower bounds follow the following general proof strategy:

1. For a given instance $(n, k, l, \theta_1, ..., \theta_n)$, consider other instances on which no algorithm can output $\{\pi_1, ..., \pi_k\}$ with high probability [1]. For example if we just change θ_{k+1} to θ_k, then no algorithm can output $\{\pi_1, ..., \pi_k\}$ with probability more than 1/2. This is because item k and item $k+1$ look the same now and thus all the algorithms will output $\{\pi_1, ..., \pi_k\}$ and $\{\pi_1, ..., \pi_{k-1}, \pi_{k+1}\}$ with the same probability in the modified instance.

[1] Recall that π_i denotes the initial label of i-th item given as the input to the algorithm, and thus the true top-k items are labeled by $\{\pi_1, ..., \pi_k\}$.

2. We then consider a well-designed distribution over these modified instances. We show that for any algorithm A with not enough comparisons, the transcript of running A on the original instance distributes very closely to the transcript of running A on the well-design distribution over modified instances.

3. Finally, since the transcript also includes the output, step 2 will tell us that if A does not use enough comparisons, then A must fail to output $\{\pi_1, ..., \pi_k\}$ with some constant probability.

8.1.4 Related Works

Rank aggregation from pairwise comparisons is an important problem in computer science, which has been widely studied under different comparison models. Most existing works focus on the non-active setting: the pairs of items for comparisons are fixed (or chosen completely at random) and the algorithm cannot adaptively choose the next pair for querying. In this non-active ranking setup, when the goal is to obtain a global ranking over all the items, Negahban et al. [159] proposed the *RankCentrality* algorithm under the popular Bradley-Terry model, which is a special case of the MNL model for pairwise comparisons. Lu and Boutilier [144] proposed a ranking algorithm under the Mallows model. Rajkumar and Agarwal [169] investigated different statistical assumptions (e.g., generalized low-noise condition) for guaranteeing to recover the true ranking. Shah et al. [175] studied the ranking aggregation under a non-parametric comparison model—strong stochastic transitivity (SST) model, and converted the ranking problem into a matrix estimation problem under shape-constraints. Most machine learning literature assumes that there is a true global ranking of items and the output of each pairwise comparison follows a probabilistic model. Another way of formulating the ranking problem is via the minimum feedback arc set problem on tournaments, which does not assume a true global ranking and aims to find a ranking that minimizes the number of inconsistent pairs. There is a vast literature on the minimum feedback arc set problem and here we omit the survey of this direction (please see [130] and references therein). Due to the increasing number of items, it is practically more useful to identify the top-k items in many internet applications. Chen and Suh [60], Jang et al. [125], and Suh et al. [179] proposed various spectral methods for top-k item identification under the BTL model or mixture of BTL models. Shah and Wainwright [177] proposed a counting-based algorithm under a general noise model including the SST model. The notion of instance optimal was originally defined and emphasized as an important concept in [96] for identifying the top-k objects from sorted lists. [57] suggested that notion "instance optimal" is necessary for rank aggregation from noisy pairwise comparisons in complicated noise models and further improved [177] under the SST noise model by proposing an algorithm that has competitive ratio $\tilde{\Theta}(\sqrt{n})$ compared to the best algorithm of each instance and proving $\tilde{\Theta}(\sqrt{n})$ is tight.

In addition to static rank aggregation, active noisy sorting and ranking problems have received a lot of attentions in recent years. For example, several works [35, 1, 124, 189] studied the active sorting problem from noisy pairwise comparisons and explored the sample complexity to approximately recover the true ranking in terms of some distance function (e.g., Kendall's tau). Chen et al. [56] proposed a Bayesian online ranking algorithm under the mixture of BTL models. Dwork et al. [92] and Ailon et al. [2] considered a related Kemeny optimization problem, where the goal is to determine the total ordering that minimizes the sum of the distances to different permutations. For top-k identification, Braverman et al. [34] initiated the study of how round complexity of active algorithms can affect the sample complexity. Szörényi et al. [182] studied the case of $k = 1$ under the BTL model. Heckel et. al. [119] investigated the active ranking under a general class of nonparametric models and also established a lower bound on the number of comparisons for parametric models. A very recent work by Mohajer and Suh [154] proposed an active algorithm for top-k identification under a general class of pairwise comparison models, where the instance difficulty is characterized by the key quantity $\min_{i\in\{1,\dots,k\}} \min_{j:j>i}(p_{ij} - 0.5)^2$. Here, p_{ij} is the probability of item i is preferred over item j. However, according to our result in Theorem 8.1.2, the obtained sample complexities in previous works are not instance optimal. We note that the lower bound result in Theorem 8.1.2 holds for algorithms even when all the values of θ_i's are known (but without the knowledge of which item corresponds to which value) and thus characterizes the difficulty of each instance. Moreover, we study the the multi-wise comparisons, which has not been explored in ranking aggregation literature but has a wide range applications.

Finally, we note that the top-k ranking problem is related to the best k arm identification in multi-armed bandit literature [44, 123, 191, 55]. However, in the latter problem, the samples are *i.i.d.* random variables rather than comparisons and the goal is to identify the top-k distributions with largest means.

8.2 Algorithm

For notational simplicity, throughout the paper we use the words w.h.p. to denote with probability $1 - 1/n^c$ for sufficiently large constant c.

8.2.1 Top-k item identification (For logarithmic l)

For $l = O(\log n)$, we can always use pairwise comparisons by losing a polylog factor as proven in Claim 8.4.2. Therefore, we only focus on the case when $l = 2$ in this section.

Before presenting the algorithm, let us first consider a graph $G = (V = [n], E)$ where each edge is labeled with either $\approx_l, \geq_l, \leq_l, >_l$ or $<_l$ (see Line 9 in Algorithm 12). Based on the labeling of edges, we give the following definition of label monotone, which will be used in Algorithm 12.

Definition 8.2.1 (Monotone)**.** *We call a path $i_1 \to i_2 \to \cdots \to i_d$ strictly label monotone if:*

1. *For every $j \in [d-1]$, the edge (i_j, i_{j+1}) is labeled with either $\approx_l, \geq_l$ or $>_l$.*

2. *There exists at least one edge (i_j, i_{j+1}) with label $>_l$.*

Moreover, we call a path "label monotone" if only property 1 holds.

Theorem 8.2.1. *For every m items with $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$, Algorithm 12, on given a random permutation of labels $\Omega = [n]$ and k, returns top-k items w.h.p. using*

$$O\left(\kappa^7 \cdot \left(k + \sum_{i=k+1}^{n} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}\right)\right)$$

total number of pairwise comparisons.

We defer the proof of Theorem 8.2.1 to Section 8.4. We only provide the pseudocode in Algorithm 12. In Algorithm 12, we note that a different letter m (instead of n) is used for denoting the set size because we will run the algorithm recursively with smaller sets. And also notice that the parameter $\kappa = \Omega(\log^2 n)$ regardless of the value of m. We also defer our result for superlogarithmic l to Section 8.4.

Below, following the intuition in section 8.1.3, line 3 of the algorithm randomly samples $\tilde{O}(m)$ paris to compare. At each iteration, the algorithm queries all pairs once. Line 9 builds a confident interval for $\tilde{\theta}_{i,j} \approx \frac{\theta_i}{\theta_i + \theta_j}$. Line 10 of the algorithm declares $\theta_j > \theta_i$ if $\tilde{\theta}_{i,j}$ is much larger than $\frac{1}{2}$ or there is a path $j = i_1 \to i_2 \to \cdots \to i_d = i$ such that for each $r \in [d-1]$, $\tilde{\theta}_{i_r, i_{r+1}} \geq \frac{1}{2}$ and there is an $r' \in [d-1]$ with $\tilde{\theta}_{i_r, i_{r+1}}$ much larger than $\frac{1}{2}$.

Algorithm 12 AlgPairwise

Parameter: $\kappa = \Omega(\log^2 n)$.
Input: A set of randomly permuted labels Ω with $|\Omega| = m$, k: number of top items.
Uniformly at random sample $s = m\kappa$ subsets $S_1, \cdots S_s$ of Ω, each of size 2. Associate these subsets with a graph $G = (\Omega, E)$, where each edge $e_u \in E$ consists of all the vertices in S_u for $u \in [s]$.
$q = 0$, $\Omega_g = \emptyset, \Omega_b = \emptyset, S = \emptyset$.
while true **do**
    $q \leftarrow q + 1$.
    Query each set 1 time, obtain in total s query results $\{R_{u,q}\}_{u\in[s]}$. ($R_{u,q}$ indicates the reported most favorable item)
    For all $u \in [s]$, for $\{i, j\} = S_u$, let $\tilde{\theta}_{i,j} = \frac{1}{q}\sum_{p\in[q]} 1_{R_{u,p}=i}$.
    For each edge $(i, j) \in E$, we label it as:

    1. $i \approx_l j$ if $\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \in \left[\frac{1}{1+4\sqrt{\frac{\kappa}{q}}}, 1 + 4\sqrt{\frac{\kappa}{q}}\right]$
    2. $i \geq_l j$ if $\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \in \left(1 + 4\sqrt{\frac{\kappa}{q}}, 1 + 32\kappa\sqrt{\frac{\kappa}{q}}\right)$
    3. $i >_l j$ if $\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \in \left[1 + 32\kappa\sqrt{\frac{\kappa}{q}}, \infty\right)$
    4. $i \leq_l j$ if $\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \in \left(\frac{1}{1+32\kappa\sqrt{\frac{\kappa}{q}}}, \frac{1}{1+4\sqrt{\frac{\kappa}{q}}}\right)$
    5. $i <_l j$ if $\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \in \left[0, \frac{1}{1+32\kappa\sqrt{\frac{\kappa}{q}}}\right]$

    For every $i, j \in [m]$, we call $j{>>}_l i$ if there exists a strictly label monotone path of length at most κ from j to i.
    For each $i \in [m]$, if there exists at least k many $j \in [m]$ such that $j{>>}_l i$, then add i to Ω_b. (Ω_b is the subset of items that we are sure not in top-k.)
    For each $i \in [m]$, if there exists at least $m - k$ many $j \in [m]$ such that $i{>>}_l j$, then add i to Ω_g. (Ω_g is the subset of items that we are sure in top-k.)
    Break if $|\Omega_g \cup \Omega_b| \geq \frac{m}{4}$.
end while
$\Omega' = \Omega - \Omega_g - \Omega_b, k' = k - |\Omega_g|$, $S = S \cup \Omega_g \cup \mathsf{AlgPairwise}(\Omega', k')$.
Return S.

8.3 Lower Bounds

Here we present lower bounds on the number of comparison used by any algorithm which identifies top-k items even when the values of preference scores $\{\theta_1, ..., \theta_n\}$ are given to the algorithm. (The algorithm just do not know which item has which θ_i). Proofs are deferred to Section 8.5.

8.3.1 Lower bounds for close weights

Theorem 8.3.1. *Assume $\theta_k > \theta_{k+1}$ and $c < 10^{-4}$. For any algorithm A (can be adaptive), if A uses $c \sum_{i:k+1 \le i, \theta_i \ge \theta_k/2} \frac{\theta_k^2}{(\theta_k - \theta_i)^2}$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $7/8$.*

Theorem 8.3.2. *Assume $\theta_k > \theta_{k+1}$ and $c < 4 \cdot 10^{-4}$. For any algorithm A (can be adaptive), if A uses $c \sum_{i:i \le k, \theta_i \le 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $7/8$.*

8.3.2 Lower bounds for arbitrary weights

Theorem 8.3.3. *Assume $c < 1/18$. For any algorithm A (can be adaptive), if A uses $c \sum_{i:i>k} \frac{\theta_i}{\theta_k}$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $7/8$.*

Theorem 8.3.4. *For any algorithm A (can be adaptive), if A uses $k/4$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $2/3$.*

Theorem 8.3.5. *Assume $c < 1/2$. For any algorithm A (can be adaptive), if A uses $\frac{cn}{l}$ comparisons of size at most l (can be 2-wise, 3-wise,...,l-wise comparisons), then A will identify the top-k items with probability at most $7/8$.*

8.3.3 Combining lower bounds

Corollary 8.3.1 (Restatement of Theorem 8.1.2)**.** *For any algorithm A (can be adaptive), suppose A uses comparisons of size at most l (can be 2-wise, 3-wise,...,l-wise comparisons). A needs*

$$\Omega\Bigg(\frac{n}{l} + k + \frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i: i \leq k, \theta_i \leq 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2} \Bigg)$$

to identify the top-k items with probability at least 7/8*.*

Proof. To prove this corollary, we just need to combine all the results in Theorem 8.3.1, Theorem 8.3.2, Theorem 8.3.3, Theorem 8.3.4 and Theorem 8.3.5. And then use the fact that if $b < a_1 + \cdots + a_5$ then there exists $i \in \{1, 2, 3, 4, 5\}$ such that $b < 5a_i$. □

8.4 Additional Algorithmic Results and Proofs

Throughout the proofs we are going to use the following claim which is a simple fact about the binomial concentration.

Claim 8.4.1 (Binomial concentration)**.** *For every $m \in \mathbb{N}^*$, every $p \in [0, 1]$, suppose $X \sim B(m, p)$, then $X \in [mp - O(\sqrt{mp \log n}), mp + O(\sqrt{mp \log n})]$ w.h.p (with high probability respect to n).*

8.4.1 Top-k item identification (For logarithmic l)

In this section, we prove Theorem 8.2.1 of Section 4.4.1.

Following Claim 8.4.1, we know that for every $(i,j) \in E$, every q,

$$\tilde{\theta}_{i,j} \in \left[\theta_{i,j} - \sqrt{\frac{\theta_{i,j}\kappa}{q}}, \theta_{i,j} + \sqrt{\frac{\theta_{i,j}\kappa}{q}}\right]$$

w.h.p. W.l.o.g, let us just focus on the case that this bound is satisfied for all $(i,j) \in E$ and every q.

We have the following Lemma about the labelling:

Lemma 8.4.1 (Label)**.** *For $q = \Omega(\kappa^3)$, we have:*

1. *if $\theta_i \geq \theta_j$, then $i \approx_l j, i \geq_l j$ or $i >_l j$.*
2. *if $\theta_i \geq \theta_j \left(1 + 128\kappa\sqrt{\frac{\kappa}{q}}\right)$, then $i >_l j$.*
3. *if $i \geq_l j$ or $i \approx_l j$, then*
$$\theta_i \geq \theta_j \left(1 - 8\sqrt{\frac{\kappa}{q}}\right)$$
4. *if $i >_l j$, then*
$$\theta_i \geq \theta_j \left(1 + 16\kappa\sqrt{\frac{\kappa}{q}}\right)$$

Proof of Lemma 8.4.1. 1. We know that for $q = \Omega(\kappa^3)$ and $\theta_i \geq \theta_j$:

$$\begin{aligned}
\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} &\geq \frac{\frac{\theta_i}{\theta_i+\theta_j} - \sqrt{\frac{\theta_i}{\theta_i+\theta_j}\cdot\frac{\kappa}{q}}}{\frac{\theta_j}{\theta_i+\theta_j} + \sqrt{\frac{\theta_j}{\theta_i+\theta_j}\cdot\frac{\kappa}{q}}} \geq \frac{\theta_i\left(1 - \sqrt{\frac{2\kappa}{q}}\right)}{\theta_j + \sqrt{\frac{2\theta_i\theta_j\kappa}{q}}} \\
&\geq \frac{1 - \sqrt{\frac{2\kappa}{q}}}{\frac{\theta_j}{\theta_i} + \sqrt{\frac{\theta_j}{\theta_i}\cdot\frac{2\kappa}{q}}} \geq \frac{1 - \sqrt{\frac{2\kappa}{q}}}{1 + \sqrt{\frac{2\kappa}{q}}} \\
&\geq \frac{1}{1 + 4\sqrt{\frac{\kappa}{q}}}
\end{aligned}$$

2. Again by $\theta_i \geq \theta_j\left(1+128\kappa\sqrt{\frac{\kappa}{q}}\right)$ and $q = \Omega(\kappa^3)$, we know that $\frac{\theta_j}{\theta_i} \leq 1 - 64\kappa\sqrt{\frac{\kappa}{q}}$, therefore, we have:

$$\frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \geq \frac{1-\sqrt{\frac{2\kappa}{q}}}{\frac{\theta_j}{\theta_i}+\sqrt{\frac{\theta_j}{\theta_i}\cdot\frac{2\kappa}{q}}} \geq \frac{1-\sqrt{\frac{2\kappa}{q}}}{\frac{\theta_j}{\theta_i}+\sqrt{\frac{2\kappa}{q}}} \geq$$

$$\frac{1-\sqrt{\frac{2\kappa}{q}}}{1-64\kappa\sqrt{\frac{\kappa}{q}}+\sqrt{\frac{2\kappa}{q}}} \geq 1+32\kappa\sqrt{\frac{\kappa}{q}}\ .$$

3. Let us suppose $\theta_i \leq \theta_j$, otherwise we already complete the proof. Now, we have:

$$\frac{\frac{\theta_i}{\theta_i+\theta_j}+\sqrt{\frac{\theta_i}{\theta_i+\theta_j}\cdot\frac{\kappa}{q}}}{\frac{\theta_j}{\theta_i+\theta_j}-\sqrt{\frac{\theta_j}{\theta_i+\theta_j}\cdot\frac{\kappa}{q}}} \geq \frac{\tilde{\theta}_{i,j}}{\tilde{\theta}_{j,i}} \geq \frac{1}{1+4\sqrt{\frac{\kappa}{q}}}$$

Which implies that

$$\frac{\theta_i+\sqrt{\frac{2\theta_i\theta_j\kappa}{q}}}{\theta_j\left(1-\sqrt{\frac{2\kappa}{q}}\right)} \geq \frac{1}{1+4\sqrt{\frac{\kappa}{q}}}$$

Therefore, by $\theta_i \leq \theta_j$, we have:

$$\frac{\frac{\theta_i}{\theta_j}+\sqrt{\frac{2\kappa}{q}}}{1-\sqrt{\frac{2\kappa}{q}}} \geq \frac{1}{1+4\sqrt{\frac{\kappa}{q}}}$$

Which implies that

$$\frac{\theta_i}{\theta_j} \geq \frac{1-\sqrt{\frac{2\kappa}{q}}}{1+4\sqrt{\frac{\kappa}{q}}} - \sqrt{\frac{2\kappa}{q}} \geq 1-8\sqrt{\frac{\kappa}{q}}$$

4. Let us suppose $\theta_i \leq 2\theta_j$, otherwise we already complete the proof. Again, we have:

$$\frac{\frac{\theta_i}{\theta_j}+\sqrt{\frac{3\kappa}{q}}}{1-\sqrt{\frac{3\kappa}{q}}} \geq 1+32\kappa\sqrt{\frac{\kappa}{q}}$$

Which implies that

$$\frac{\theta_i}{\theta_j} \geq \left(1 - \sqrt{\frac{3\kappa}{q}}\right)\left(1 + 32\kappa\sqrt{\frac{\kappa}{q}}\right) - \sqrt{\frac{2\kappa}{q}} \geq$$
$$1 + 16\kappa\sqrt{\frac{\kappa}{q}}$$

□

Above, the Lemma 8.4.1 implies that w.h.p. the labelling of each edge (i, j) is consistent with the order of θ_i, θ_j. Now, the algorithm will declare $i >>_l j$ if there exists strictly label monotone path from i to j. Using the Lemma above we can show that if such path exists, then $\theta_i > \theta_j$. To show the other direction that such paths exists when $\theta_j > \theta_i$, we first consider the following graph Lemma that gives the exists of monotone path in random graph $G(m, p)$.

Lemma 8.4.2 (Graph Path)**.** *For every $m \leq n$, every random graph $G(m, p)$ on vertices $V = [m]$, if $p \geq \frac{\kappa}{m}$, then w.h.p. For every $i, j \in [m]$ with $j \geq i + \frac{m}{4}$, there exists a path $i = i_1 \to i_2 \to \cdots i_d = j$ such that*

1. *$d \leq \kappa$.*

2. *$i_r \leq i_{r+1}$ for every $r \in [d-1]$.*

We call such a path a monotone path from i to j.

Proof of Lemma 8.4.2. It is sufficient to consider the case when $m = \Omega(\sqrt{\kappa})$, otherwise w.h.p. the graph is a complete graph and theorem is automatically true.

We consider a sequential way of generating $G(m, p)$: At each time $t = 1, 2, \cdots, m$, a vertex t arrives and there exists an edge between t and each $t' \in [t-1]$ with probability p. Let us consider a fixed $i \leq \frac{3}{4}m$ and $j \geq i + \frac{m}{4}$. Let $\tau = \frac{\sqrt{\kappa}}{32} = \Omega(\log n)$. We will divide the set $\{i, i+1, \cdots, j-1\}$ into τ subsets $H_1, \cdots, H_\tau$ such that

$$H_r = \{i + (r-1)\frac{j-i}{\tau}, i + (r-1)\frac{j-i}{\tau} + 1, \cdots,$$
$$i + r\frac{j-i}{\tau} - 1\}$$

Since $m = \Omega(\sqrt{\kappa})$ we know that $|H_r| \geq 1$.

Let us define the random variable Y_r, X_r, Z_v as:

$$Y_r = \text{the set of all } v \in H_r \text{ such that there exists a}$$
$$\text{monotone path from } i \text{ to } v \text{ of length at most } r$$

and $X_r = |Y_r|$.

For each $v \in H_{r+1}$, we define

$$Z_v = 1_{\text{there is an edge between } v \text{ and at least one vertex in } Y_r}$$

Clearly, $X_1 \geq 1$ and each Z_v is i.i.d. random variable in $\{0, 1\}$ with $\Pr[Z_v = 0 \mid X_r] = (1-p)^{X_r}$. On the other hand, by definition,

$$X_{r+1} \geq \sum_{v \in H_{r+1}} Z_v$$

We consider two cases:

1. $X_r \geq \frac{1}{p}$, then $\Pr[Z_v = 1] \geq \frac{1}{4}$.

2. $X_t < \frac{1}{p}$, then by $(1-p)^x \leq 1 - \frac{xp}{2}$ for $x < 1/p$, we have $\Pr[Z_v = 1] \geq \frac{pX_r}{2}$.

Consider a fixed X_r and for each $v \in H_{r+1}$, let Z_v be the random variable. By standard Chernoff bound, we have:

1. If $X_r \geq \frac{1}{p}$, then w.h.p. $X_{r+1} \geq \frac{j-i}{4\tau} \geq \frac{m}{16\tau}$.

2. $1 \le X_r < \frac{1}{p}$, then w.h.p. $X_{r+1} \ge \frac{(j-i)pX_r}{2\tau} - \sqrt{\tau \frac{(j-i)pX_r}{2\tau}}$.

Recall that $p \ge \frac{\kappa}{m}$ and $j - i \ge \frac{m}{4}$, therefore,

$$\sqrt{\tau \frac{(j-i)pX_r}{2\tau}} \le \frac{(j-i)pX_r}{4\tau}, \quad \frac{(j-i)pX_r}{4\tau} \ge 2X_r$$

Which implies that w.h.p. $X_{r+1} \ge 2X_r$.

Putting everything together, we know that for $\tau = \Omega(\log n)$, w.h.p. $X_\tau \ge \frac{m}{16\tau}$. Therefore, condition on this event, by

$$\begin{aligned}
&\Pr\left[\text{there is an edge between } j \text{ and } Y_\tau \middle| X_\tau \ge \frac{m}{16\tau}\right] \\
&= 1 - (1-p)^{\frac{m}{16\tau}} \ge 1 - \left(1 - \frac{\kappa}{m}\right)^{\frac{m}{16\tau}} \\
&\ge 1 - \left(1 - \frac{1024\tau^2}{m}\right)^{\frac{m}{16\tau}} \ge 1 - \frac{1}{n^{\Omega(1)}}
\end{aligned}$$

We complete the proof.

□

Having this Lemma, we can present the main Lemma above the algorithm:

Lemma 8.4.3 (Main 3)**.** *Suppose $q = \Omega(\kappa^3)$, then w.h.p. the following holds:*

1. $\Omega_g \subseteq [k]$, $\Omega_b \cap [k] = \emptyset$.
2. *If* $k \le \frac{m}{2}$ *and* $q = \Omega\left(\frac{\kappa^5}{m} \cdot \sum_{i=k+1}^{m} \frac{\theta_k^2}{(\theta_k - \theta_i)^2}\right)$, *then* $|\Omega_b| \ge \frac{m}{4}$.
3. *If* $k > \frac{m}{2}$ *and* $q = \Omega\left(\frac{\kappa^5}{m} \cdot \left(k + \sum_{i=1}^{k} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}\right)\right)$, *then* $|\Omega_g| \ge \frac{m}{4}$.

Since the algorithm terminates within $O(\kappa)$ recursions, moreover, in each recursion, the algorithm makes at most $\kappa m q$ queries. Therefore, Lemma 8.4.3 implies that the algorithm runs in total queries:

$$O\left(\kappa^7 \cdot \left(k + \sum_{i=k+1}^{m} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}\right)\right)$$

Proof of Lemma 8.4.3. 1. It suffices to show that if $i{>>}_l j$, then $\theta_i \geq \theta_j$. To see this, consider a strictly label monotone path $i = i_1 \to i_2 \to \cdots \to i_d = j$ with length $d \leq \kappa$. By Lemma 8.4.1, we know that for every $r \in [d-1]$, we have: $\theta_{i_r} \geq \theta_{i_{r+1}} \left(1 - 8\sqrt{\frac{\kappa}{q}}\right)$. Moreover, there exists an $r' \in [d-1]$ such that $\theta_{i_{r'}} \geq \theta_{i_{r'+1}} \left(1 + 16\kappa\sqrt{\frac{\kappa}{q}}\right)$. Multiply every thing together, we know that

$$\theta_i \geq \theta_j \left(1 - 8\sqrt{\frac{\kappa}{q}}\right)^{\kappa-1} \left(1 + 16\kappa\sqrt{\frac{\kappa}{q}}\right) \geq \theta_j$$

2. Let us denote the set $H = \{\frac{3}{4}m + 1, \frac{3}{4}m + 2, \cdots m\}$, we will prove that $H \subseteq \Omega_b$. Consider one $j \in H$, by Lemma 8.4.2, w.h.p. for every $i \in [k]$, there exists a path $i = i_1 \to i_2 \to \cdots i_d = j$ of length at most κ such that $\theta_{i_r} \geq \theta_{i_{r+1}}$ for every $r \in [d-1]$. Now, by Lemma 8.4.1, we know that this path is label monotone. We just need to show that this path is strictly label monotone. To see this, we know that there exists one $r' \in [d-1]$ such that

$$\theta_{i_{r'}} \geq \theta_{i_{r'+1}} \left(\frac{\theta_i}{\theta_j}\right)^{1/\kappa}$$

Let $\nu = \frac{1}{m-k}\sum_{i=k+1}^{m} \frac{\theta_k^2}{(\theta_k - \theta_i)^2}$. Now, since $k \leq \frac{m}{2}$, we can apply Markov inequality and conclude that for this $j \in H$ and $i \in [k]$, $\frac{\theta_i^2}{(\theta_i - \theta_j)^2} \leq \frac{\theta_k^2}{(\theta_k - \theta_j)^2} \leq 2\nu$. Which implies that

$$\frac{\theta_i}{\theta_j} \geq \frac{1}{1 - \sqrt{\frac{1}{2\nu}}} \geq 1 + \sqrt{\frac{1}{2\nu}}$$

For $q = \Omega(\kappa^5 \nu)$, we know that

$$\left(\frac{\theta_i}{\theta_j}\right)^{1/\kappa} \geq \left(1 + 64\kappa^2\sqrt{\frac{\kappa}{q}}\right)^{1/\kappa} \geq \left(1 + 32\kappa\sqrt{\frac{\kappa}{q}}\right)$$

Therefore, $\theta_{i_{r'}} \geq \theta_{i_{r'+1}} \left(\frac{\theta_i}{\theta_j}\right)^{1/\kappa} \geq \left(1 + 32\kappa\sqrt{\frac{\kappa}{q}}\right)$. By definition, we shall label $i_{r'} > i_{r'+1}$ and thus $i{>>}_l j$.

3. It can be shown with exactly the same calculation as 2 with $H = \left\{1, 2, \cdots \frac{1}{4}m\right\}$ and apply Markov inequality on

$$\nu = \frac{1}{k}\sum_{i=1}^{k}\frac{\theta_i^2}{(\theta_{k+1}-\theta_i)^2} = \frac{1}{k}O\left(k + \sum_{i=1}^{k}\frac{\theta_{k+1}^2}{(\theta_{k+1}-\theta_i)^2}\right)$$

□

8.4.2 Top-k item identification (For super logarithmic l)

Before presenting the algorithm, we first argue about which case using bigger l is unnecessary. We have the following Claim:

Claim 8.4.2 (Bigger l)**.** *For every $l \leq m \leq n$, we have:*

$$\begin{aligned}
&k + \sum_{i=k+1}^{m}\frac{\theta_k^2}{(\theta_k-\theta_i)^2} + \sum_{i=1}^{k}\frac{\theta_{k+1}^2}{(\theta_{k+1}-\theta_i)^2} \\
&\leq \left(\frac{m}{l} + k + \frac{\sum_{i\geq k+1}\theta_i}{\theta_k} + \sum_{i\geq k+1, \theta_i \geq \frac{\theta_k}{2}}\frac{\theta_k^2}{(\theta_k-\theta_i)^2} + \right. \\
&\left.\sum_{i=1}^{k}\frac{\theta_{k+1}^2}{(\theta_{k+1}-\theta_i)^2}\right) + 4m
\end{aligned}$$

Therefore, as long as we can show one of the following:

1. $k = \Omega(m)$.

2. $\frac{\sum_{i\geq k+1}\theta_i}{\theta_k} = \Omega(m)$.

We can just use the algorithm for $l = 2$. Otherwise, we shall consider larger l, we will directly considering the case when $l = \Omega(\log n)$. Before giving the algorithm, it is convenient to first consider the following query procedure: For a fixed Q, do:

Algorithm 13 BasicQuery

Parameter: $\kappa = \Omega(\log^2 n)$

Input: Ω: set of items with $|\Omega| = m$, k: number of top items to find. l: size of the subset to query.

Uniformly at random sample $s = \frac{m\kappa}{l}$ subsets $S_1, \cdots S_s$ of Ω, each of size l. Associate these subsets with a hypergraph $G = (\Omega, E)$, where each edge $e_u \in E$ is consists of all the vertices in S_u for $u \in [s]$.

Query each set Q time, obtain in total sQ query results $\{R_{u,q}\}_{u\in[s],q\in[Q]}$.

For a fixed $q \leq Q$, let us consider a random variable $\tilde{\theta}_{i,S_u} \in [0,1]$ defined as

$$\tilde{\theta}_{i,S_u} = \frac{1}{q}\sum_{r\in[q]} 1_{R_{u,r}=i}$$

For each i, u such that $i \in S_u$, let us define 0-1 valued function $1_{i,u,\alpha,\beta,\gamma}$ such that $1_{i,u,\alpha,\beta,\gamma} = 1$ if and only if all the following conditions hold:

1. $\tilde{\theta}_{i,S_u} \geq \frac{\alpha}{q}$.

2. There exists at least γl many of the $j \in S_u$ such that $\tilde{\theta}_{j,S_u} \leq \beta\tilde{\theta}_{i,S_u}$.

We also consider the random variable $X_{i,u,\alpha,\beta,\gamma}$ associated with this function, where the randomness is taken over the uniformly at random choice of S_u conditional on $i \in S_u$, and the randomness of the outcome of the queries.

We prove the following main Lemma:

Lemma 8.4.4 (Indicator)**.** *Let* $\gamma \in \left[\frac{1}{32}, \frac{1}{2}\right], \beta \in (0, 32]$, $\alpha = \Omega(\kappa)$. *For every* $i \in [m]$, *the following holds:*

1. *If* $q = \Omega\left(\alpha + \frac{2\alpha l \sum_{j\in[m]}\theta_j}{m\theta_i}\right)$ *and* $\theta_i \geq 2\beta\theta_{(1-2\gamma)m}$, *then*

$$\Pr[X_{i,u,\alpha,\beta,\gamma} = 1] \geq \frac{15}{16}$$

2. *For every* q, *if* $\theta_i \leq \frac{\beta}{2}\theta_{(1-\gamma)m}$, *then*

$$\Pr[X_{i,u,\alpha,\beta,\gamma} = 1] \leq \frac{9}{16}$$

Proof of Lemma 8.4.4. 1. We first bound the probability that $\tilde{\theta}_{i,S_u} \geq \frac{\alpha}{q}$. By

$$\tilde{\theta}_{i,S_u} \in \left[\theta_{i,S_u} - \sqrt{\frac{\theta_{i,S_u}\kappa}{q}}, \theta_{i,S_u} + \sqrt{\frac{\theta_{i,S_u}\kappa}{q}}\right]$$

we know that

$$\theta_{i,S_u} \geq \frac{2\alpha}{q} \implies \tilde{\theta}_{i,S_u} \geq \frac{\alpha}{q}$$

To lower bound this probability, we just need to consider the probability that $\theta_{i,S_u} < \frac{2\alpha}{q}$. We apply Markov inequality and have that:

$$\Pr\left[\theta_{i,S_u} < \frac{2\alpha}{q}\right] = \Pr\left[\frac{2\alpha}{q\theta_{i,S_u}} > 1\right] < \frac{\mathrm{E}\left[\frac{2\alpha}{q\theta_{i,S_u}}\right]}{1}$$

Notice that

$$\frac{2\alpha}{q\theta_{i,S_u}} = \frac{2\alpha \sum_{j\in S_u} \theta_j}{\theta_i q}$$

Therefore,

$$\mathrm{E}\left[\frac{2\alpha}{q\theta_{i,S_u}}\right] = \frac{2\alpha\,\mathrm{E}[\sum_{j\in S_u}]\theta_j}{\theta_i q} \leq \frac{2\alpha}{q} + \frac{2\alpha l \sum_{j\in[m]} \theta_j}{m\theta_i q}$$
$$\leq \frac{1}{64}$$

Putting together we obtain

$$\Pr\left[\tilde{\theta}_{i,S_u} < \frac{\alpha}{q}\right] \leq \Pr\left[\theta_{i,S_u} < \frac{2\alpha}{q}\right] < \frac{1}{64}$$

Now we can move to the second condition. For now, suppose $\tilde{\theta}_{i,S_u} \geq \frac{\alpha}{q}$ holds, we then know that

$$\tilde{\theta}_{i,S_u} \in \left[\frac{31}{32}\theta_{i,S_u}, \frac{33}{32}\theta_{i,S_u}\right]$$

Therefore, $\theta_i \geq 2\beta\theta_{(1-2\gamma)m}$ implies that for every $j \in H = \{(1-2\gamma)m, (1-2\gamma)m+1, \cdots m\}$ with $j \in S_u$, we have:

$$\begin{aligned}
\tilde{\theta}_{j,S_u} &\leq \theta_{j,S_u} + \sqrt{\frac{\theta_{j,S_u}\kappa}{q}} \leq \theta_{j,S_u} + \sqrt{\frac{\theta_{i,S_u}\kappa}{q}} \\
&\leq \theta_{j,S_u} + \frac{1}{128}\theta_{i,S_u} \leq \frac{\theta_{i,S_u}}{2\beta} + \frac{1}{128}\theta_{i,S_u} \\
&\leq \frac{3\theta_{i,S_u}}{4\beta} \quad \text{for } \beta \leq 32 \\
&\leq \frac{\tilde{\theta}_{i,S_u}}{\beta}
\end{aligned}$$

Since $|H| = 2\gamma m$, we know that for $l = \Omega(\log n)$, $\Pr[|H \cap S_u| < \gamma l] \leq \frac{1}{64}$. Therefore,

$$\begin{aligned}
&\Pr[X_{i,u,\alpha,\beta,\gamma} = 1] \geq \\
&1 - \Pr\left[\tilde{\theta}_{i,S_u} < \frac{\alpha}{q}\right] - \Pr[|H \cap S_u| < \gamma l] \geq \frac{15}{16}
\end{aligned}$$

2. The proof follows from the same calculation. Notice that this time we already have $X_{i,u,\alpha,\beta,\gamma} = 1 \implies \tilde{\theta}_{i,S_u} \geq \frac{\alpha}{q}$.

□

For fixed $\alpha = \Omega(\kappa)$, every $\beta \in (0, 32], \gamma \in \left[\frac{1}{32}, \frac{1}{2}\right]$ and every $\tau \in \left[\frac{3}{4}, \frac{7}{8}\right]$, we consider set

$$\Omega_{\beta,\gamma,\tau} = \left\{ i \in [m] \middle| \sum_{u: u \in [s], i \in S_u} X_{i,u,\alpha,\beta,\gamma} \geq \tau \deg(i) \right\}$$

We also have the following Corollary of Lemma 8.4.4:

Corollary 8.4.1. *1. For every $i, j \in [m]$ with $\theta_i \geq \theta_j$, every $\tau \in \left[\frac{3}{4} \times \frac{33}{32}, \frac{7}{8}\right]$, w.h.p. $j \in \Omega_{\beta,\gamma,\tau} \implies i \in \Omega_{\beta,\gamma,\frac{32}{33}\tau}$.*

2. For every $i \in [m]$, if $\theta_i \geq 2\beta\theta_{(1-2\gamma)m}$ and $q = \Omega\left(\alpha + \frac{2\alpha l \sum_{j \in [m]} \theta_j}{m\theta_i}\right)$, then w.h.p. $i \in \Omega_{\beta,\gamma,\tau}$.

3. For every q, if $i \in \Omega_{\beta,\gamma,\tau}$, then w.h.p. $\theta_i \geq \frac{\beta}{2}\theta_{(1-\gamma)m}$.

Having this Corollary, we can do the following algorithm that selects all the $\theta_i \geq 32\max\{\theta_k, \theta_{\frac{3}{4}m}\}$ and removes most of the $\theta_j \leq \frac{1}{4}\theta_k$:

Algorithm 14 AlgMulti-wise

Parameter: $\kappa = O(\log^2 n)$.
Input: Ω: set of items, k: number of top items to find.
Output: S: set of top items. Ω': set of remaining items.
Initialization: $S = \emptyset, \Omega' = \Omega$, $m = |\Omega|$.
Call BasicQuery to obtain $\{\tilde{\theta}_{i,S_u}\}_{i\in[n],u\in[s]}$.
if $k \leq \frac{1}{2}m$ **then**
    if $1 \leq |\Omega_{32,\frac{1}{4},\frac{13}{16}}|$ and $|\Omega_{4,\frac{1}{16},\frac{13}{16}}| < k$ **then**
        $S_1 = \Omega_{4,\frac{1}{16},\frac{7}{8}}$, $\Omega'' = \Omega - S_1$, $(S', \Omega') =$ AlgMulti-wise$(\Omega'', k - |S_1|, R)$, $S = S \cup S_1 \cup S''$.
        Notice that we pick those numbers so $\frac{7}{8} \geq \frac{33}{32}\cdot\frac{13}{16} \geq \left(\frac{33}{32}\right)^2\cdot\frac{3}{4}$.
    else if $|\Omega_{4,\frac{1}{16},\frac{13}{16}}| \geq k$ **then**
        $\Omega'' = \Omega_{4,\frac{1}{16},\frac{3}{4}}$, $(S', \Omega') =$ AlgMulti-wise(Ω'', k, R), $S = S \cup S'$.
    end if
end if
Return S, Ω'.

We have the following lemma.

Lemma 8.4.5. *For every m, every $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_m$, every $k \leq m$, every $l \leq m$, Algorithm 14, on given a random permutation of $\Omega = [m], k$ satisfies:*

1. *Output set (S, Ω') of the algorithm satisfies $S \subseteq [k]$.*
2. *If $Q = \tilde{\Omega}\left(1 + \frac{l\left(k + \sum_{j\geq k}\theta_j\right)}{m\theta_k}\right)$, then the algorithm returns in $O(\log m)$ many recursion calls, and after the algorithm, let us for simplicity still denote $[|\Omega'|] = \Omega'$ with $\theta_1 \geq \theta_2 \geq \cdots \theta_{|\Omega'|}$ and $k' = k - |S|$, then either*
   - *(a) For every $i \in \Omega'$, $\theta_i \leq 512\theta_{\frac{7}{8}|\Omega'|}$.*
   - *(b) Or $k' \geq \frac{1}{2}|\Omega'|$.*

Proof of the main theorem. After running this algorithm, we can simply apply the algorithm for $l = 2$ (By Claim 8.4.2), since one of the following is true:

1. $\frac{\sum_{i\geq k'+1}\theta_i}{\theta_{k'}} \geq \frac{3}{8} \times \frac{1}{512}|\Omega'|$.

2. $k' > \frac{1}{2}|\Omega'|$.

Therefore, putting everything together, we can get the top k items in total number of queries:

$$\tilde{O}\Bigg(\frac{m}{l} + k + \frac{\sum_{i\geq k+1}\theta_i}{\theta_k} + \sum_{i\geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}\Bigg)$$

□

Now, it just remains to prove this Lemma:

Proof. We first prove the correctness: $S \subseteq [k]$. We have the following observations:

1. If $|\Omega_{4,\frac{1}{16},\frac{13}{16}}| \geq k$, then there must be $i \in \Omega_{4,\frac{1}{16},\frac{13}{16}}$ with $\theta_i \leq \theta_k$. By Corollary 8.4.1, since $\frac{13}{16} \geq \frac{33}{32} \cdot \frac{3}{4}$, we know that $[k] \subseteq \Omega_{4,\frac{1}{16},\frac{3}{4}}$.

2. If there exists $j \in \Omega_{4,\frac{1}{16},\frac{7}{8}}$ such that $j \notin [k]$, then by $\theta_j \leq \theta_k$, apply Corollary 8.4.1 with $\frac{7}{8} \geq \frac{33}{32} \cdot \frac{13}{16}$, we know that $[k] \subseteq \Omega_{4,\frac{1}{16},\frac{13}{16}}$, which implies $|\Omega_{4,\frac{1}{16},\frac{13}{16}}| \geq k$. Therefore, we will not include any item that is not top k to S when recursing from Line 7.

These two observations immediately imply $S \subseteq [k]$.

Now, we will show that for sufficiently large Q, either of the two conditions hold:

1. For every $i \in \Omega'$, $\theta_i \leq 512\theta_{\frac{7}{8}|\Omega'|}$.

2. Or $k' \geq \frac{1}{2}|\Omega'|$.

Let us for notation simplicity drop the $'$ here. Clearly, we just need to consider the case when $k < \frac{1}{2}m$, otherwise, the algorithm will just terminate and the second condition is true. We will first prove that $\theta_i \leq 64\theta_k$ and then we prove that $\theta_k \leq 8\theta_{\frac{7}{8}m}$.

1. To prove $\theta_i \leq 64\theta_k$, we suppose on the contrary that $\theta_1 > 64\theta_k$. Apply Corollary 8.4.1 with $q = Q = \Omega\left(\alpha + \frac{2\alpha l(\sum_{j\geq 1}\theta_j)}{m\theta_1}\right)$, we have that $1 \in \Omega_{32,\frac{1}{16},\frac{3}{4}}$, which implies that $|\Omega_{32,\frac{1}{4},\frac{13}{16}}| > 0$, so the algorithm won't terminate, contradict.

2. Now, we need to show that $\theta_k \leq 8\theta_{\frac{7}{8}n}$. We also on the contrary suppose that $\theta_k > 8\theta_{\frac{7}{8}m}$. Since the algorithm termniates, by the previous claim, we know that in the last recursion, it must be the case that $\theta_1 \leq 64\theta_k$. Which implies that

$$\begin{aligned} Q &= \Omega\left(\alpha + \frac{2\alpha l\left(k\theta_k + \sum_{j\geq k}\theta_j\right)}{m\theta_k}\right) \\ &= \Omega\left(\alpha + \frac{2\alpha l\left(\sum_{j\geq 1}\theta_j\right)}{m\theta_k}\right) \end{aligned}$$

   Therefore, if $\theta_k > 8\theta_{\frac{7}{8}m}$, then by Corollary 8.4.1 we know that $[k] \subseteq \Omega_{4,\frac{1}{16},\frac{3}{4}}$, so the algorithm won't terminate.

Finally, we consider about the total number of recursions. Clearly, if the algorithm recurses through the second case, then $|\Omega'| \leq \frac{15}{16}|\Omega|$. If the algorithm recurses through the first case, then by Corollary 8.4.1, it must be the case that

$$\theta_1 \geq 16\theta_{\frac{7}{8}m}$$

Which implies that for all i with $\theta_i \geq \frac{\theta_1}{2} \geq 8\theta_{\frac{7}{8}m}$, $i \in \Omega_{4,\frac{1}{16},\frac{7}{8}}$.

Therefore, the total number of recursions of the algorithm is bounded by $O(\log m)$. So the total number of queries of the algorithm is:

$$\begin{aligned} &O\left(\alpha + \frac{2\alpha l\left(k\theta_k + \sum_{j\geq k}\theta_j\right)}{m\theta_k}\right) \times \frac{\kappa m}{l} \times O(\log m) \\ &= \tilde{O}\left(\frac{m}{l} + k + \frac{\sum_{j\geq k}\theta_j}{\theta_k}\right) \end{aligned}$$

□

Remark: How to obtain the value Q: In the proof above we assumed that we have an aprior estimation of the value of Q. We can replace this assumption by initially setting Q to be $Q = Q_0 = 1$, and run algorithm 14 with Q_0 queries and then run the algorithm with pairwise comparision. Once the later algorithm requires more than $Q_0 \times \frac{n}{l}$ queries, then we stop it, set $Q_1 = 2Q_0$ and repeat this procedure with $Q = Q_1$. We keep on repeating this for $Q_2 = 2Q_1, Q_3 = 2Q_2, \cdots$ until the later algorithm requires less than $Q_i \times \frac{n}{l}$ queries.

By the Lemma we just proved, the output of the algorithm is correct for every Q. Moreover, if

$$Q \times \frac{n}{l} = \tilde{\Omega}\Bigg(\frac{n}{l} + k + \frac{\sum_{i \geq k+1} \theta_i}{\theta_k} + \sum_{i \geq k+1, \theta_i \geq \frac{\theta_k}{2}} \frac{\theta_k^2}{(\theta_k - \theta_i)^2} + \sum_{i=1}^{k} \frac{\theta_{k+1}^2}{(\theta_{k+1} - \theta_i)^2}\Bigg)$$

Then this process will terminate, and the total query complexity is then bounded by $\tilde{O}\left(Q \times \frac{n}{l}\right)$.

8.5 Proofs of Lower Bounds

8.5.1 Lower bounds for close weights

Theorem 8.5.1 (Restatement of Theorem 8.3.1)**.** *Assume $\theta_k > \theta_{k+1}$ and $c < 10^{-4}$. For any algorithm A (can be adaptive), if A uses $c \sum_{i: k+1 \leq i, \theta_i \geq \theta_k/2} \frac{\theta_k^2}{(\theta_k - \theta_i)^2}$ comparisons of any size (can be l-wise comparison for $2 \leq l \leq n$), then A will identify the top-k items with probability at most $7/8$.*

Proof. For notation convenience, we set $w_i = \frac{\theta_k^2}{(\theta_k - \theta_i)^2}$ for i such that $i \geq k+1$ and $\theta_i \geq \theta_k/2$. For other i, we set $w_i = 0$. We also set $W = \sum_{i=1}^{n} w_i$. Then we have $T = cW$.

First of all, we can assume A is deterministic. This is because if A is randomized, we can fix the randomness string which makes A achieves the highest successful probability.

Let $S = (S_1, ..., S_T)$ be the history of algorithm. Each S_t is the comparison result of round t. Notice that since A is deterministic, with $S_1, ..., S_t$, we can determine the labels of items A want to compare in round $t+1$ even when A is adaptive. So there is no point to put the labels of compared items in the history. So we only put the comparison result in the history, i.e S_t is a number in $[n]$ and S is a length-T string of numbers in $[n]$.

Again since A is deterministic, the label A outputs is just a deterministic function of S, we use $A(S)$ to denote it. A outputs correctly if A outputs the label of the top-k items, i.e. $A(S) = \{\pi_1, ..., \pi_k\}$.

We use $p(S, \pi)$ to denote the probability that the items are labeled as π and A has history S. Now consider the case when we set θ_i equals to θ_k for $i \geq k+1$. In this case the probability of $A(S) = \{\pi_1, ..., \pi_k\}$ should be at most $1/2$ as item k and item i have the same weight. We use $p_i(S, \pi)$ to denote the probability that the items are labeled as π and A has history S when θ_i is changed to θ_k.

Now we prove the following lemma that gives the connection between $p(S, \pi)$ and $p_i(S, \pi)$.

Lemma 8.5.1. *Consider p as a distribution over (π, S). For all $c_1 > 0$, we have*

$$\Pr_{(\pi,S)\sim p}\left[\left(\sum_{i=1}^{n} \frac{w_i}{W} \ln \frac{p_i(S,\pi)}{p(S,\pi)}\right) \leq -c_1\right]$$
$$\leq \exp\left(-\frac{(c_1-4c)^2}{72c}\right).$$

Proof. Define random variable Z_t to be the following for $t = 1, ..., T$ when (π, S) is sampled from distribution p:

$$Z_t = \sum_{i=1}^{n} \frac{w_i}{W} \ln \frac{p_i(S_1...S_t,\pi)}{p(S_1...S_t,\pi)}.$$

We have

$$Z_T = \sum_{i=1}^{n} \frac{w_i}{W} \ln \frac{p_i(S,\pi)}{p(S,\pi)}.$$

Now we want to show that sequence $0, Z_1+\frac{4}{W}, ..., Z_t+\frac{4t}{W}, ..., Z_T+\frac{4T}{W}$ forms a supermartingale.

Suppose in round t, given $S_1, ..., S_{t-1}$ and π, Algorithm A compares items in set U_t. Let $\theta_{-i} = \sum_{j\in U_t, j\neq i} \theta_i$. Then we have, with probability $\theta_i/(\theta_i + \theta_{-i})$,

$$\begin{aligned} Z_t - Z_{t-1} = & \frac{w_i}{W} \ln\left(1 + \frac{(\theta_k - \theta_i)\theta_{-i}}{(\theta_k + \theta_{-i})\theta_i}\right) \\ & + \sum_{j\in U_t, j\neq i} \frac{w_j}{W} \ln\left(1 - \frac{\theta_k - \theta_j}{\theta_k + \theta_{-j}}\right) \end{aligned}$$

Here are two simple facts about ln. For $0 \leq x \leq 1$, $\ln(1+x) \geq x - x^2$. For $0 \leq x \leq 1/2$, $\ln(1-x) \geq -x - x^2$. It's easy to check that for i such that $w_i > 0$, we have $\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_k+\theta_{-i})\theta_i} \leq 1$ and $\frac{\theta_k-\theta_i}{\theta_k+\theta_{-i}} \leq 1/2$. Therefore, by these two facts, for i such that $w_i > 0$, we have

$$\begin{aligned} & \frac{\theta_i}{\theta_i + \theta_{-i}} w_i \ln\left(1 + \frac{(\theta_k - \theta_i)\theta_{-i}}{(\theta_k + \theta_{-i})\theta_i}\right) \\ & + \frac{\theta_{-i}}{\theta_i + \theta_{-i}} w_i \ln\left(1 - \frac{\theta_k - \theta_i}{\theta_k + \theta_{-i}}\right) \\ \geq \; & \frac{w_i}{\theta_i + \theta_{-i}} \Big(\frac{(\theta_k - \theta_i)\theta_{-i}}{\theta_k + \theta_{-i}} - \frac{(\theta_k - \theta_i)^2\theta_{-i}^2}{(\theta_k + \theta_{-i})^2\theta_i} \\ & - \frac{(\theta_k - \theta_i)\theta_{-i}}{\theta_k + \theta_{-i}} - \frac{(\theta_k - \theta_i)^2\theta_{-i}}{(\theta_k + \theta_{-i})^2} \Big) \\ = \; & -w_i \frac{(\theta_k - \theta_i)^2\theta_{-i}}{(\theta_k + \theta_{-i})^2\theta_i} = -\frac{\theta_{-i}\theta_k^2}{(\theta_k + \theta_{-i})^2\theta_i} \\ \geq \; & -\frac{2\theta_{-i}\theta_k}{(\theta_k + \theta_{-i})^2} \geq -\frac{4\theta_i}{\theta_k + \theta_{-i}} \geq -\frac{4\theta_i}{\theta_i + \theta_{-i}}. \end{aligned}$$

Therefore we have for all t and $S_1, ..., S_{t-1}$,

$$\mathrm{E}[Z_t - Z_{t-1} | S_1, ..., S_{t-1}] \geq -\sum_{i\in U_t} \frac{4\theta_i}{W(\theta_i + \theta_{-i})} \geq -\frac{4}{W}.$$

As $Z_1, ..., Z_{t-1}$ can be determined by $S_1, ..., S_{t-1}$, we have for all t and $Z_1, ..., Z_{t-1}$,

$$\mathrm{E}\Big[\left(Z_t + \frac{4t}{W} \right) - \left(Z_{t-1} + \frac{4(t-1)}{W} \right) | Z_1 - \frac{4}{W}, ..., \\ Z_{t-1} - \frac{4(t-1)}{W} \Big] \geq 0.$$

Therefore sequence $0, Z_1 + \frac{4}{W}, ..., Z_t + \frac{4t}{W}, ..., Z_T + \frac{4T}{W}$ forms a supermartingale.

Now we want to bound $|Z_t - Z_{t-1}|$. We know that for $0 \leq x \leq 1$, $|\ln(1+x)| \leq x$ and for $0 \leq x \leq 1/2$, $|\ln(1-x)| \leq 2|x|$. Therefore for i such that $w_i > 0$,

$$|\frac{w_i}{W} \ln \left(1 + \frac{(\theta_k - \theta_i)\theta_{-i}}{(\theta_k + \theta_{-i})\theta_i} \right) | \leq \frac{w_i}{W} \cdot \frac{(\theta_k - \theta_i)\theta_{-i}}{(\theta_k + \theta_{-i})\theta_i} \\ \leq \frac{\theta_k^2 \theta_{-i}}{W\theta_i(\theta_k - \theta_i)(\theta_k + \theta_{-i})} \leq \frac{2\theta_k}{W(\theta_k - \theta_{k+1})}$$

and

$$|\frac{w_i}{W} \ln \left(1 - \frac{\theta_k - \theta_i}{\theta_k + \theta_{-i}} \right) | \leq \frac{w_i}{W} \cdot \frac{2(\theta_k - \theta_i)}{\theta_k + \theta_{-i}} \\ = \frac{2\theta_k^2}{W(\theta_k - \theta_i)(\theta_k + \theta_{-i})} \leq \frac{4\theta_k}{W(\theta_k - \theta_{k+1})} \cdot \frac{\theta_i}{\theta_i + \theta_{-i}}.$$

Therefore, we get

$$|Z_t - Z_{t-1}| \leq \frac{2\theta_k}{W(\theta_k - \theta_{k+1})} + \sum_{i \in U_t} \frac{4\theta_k}{W(\theta_k - \theta_{k+1})} \cdot \\ \frac{\theta_i}{\theta_i + \theta_{-i}} \leq \frac{6\theta_k}{W(\theta_k - \theta_{k+1})}.$$

Also notice that

$$\left(\frac{\theta_k}{(\theta_k - \theta_{k+1})} \right)^2 \leq w_{k+1} \leq W.$$

Now by Azuma's inequality, we have

$$
\begin{aligned}
& \Pr_{(\pi,S)\sim p}\left[Z_T \le -c_1\right] \\
\le\ & \exp\left(-\frac{(c_1 - \frac{4T}{W})^2}{2T(\frac{6\theta_k}{W(\theta_k-\theta_{k+1})})^2}\right) \\
=\ & \exp\left(-\frac{(c_1-4c)^2(\theta_k-\theta_{k+1})^2 W}{72\cdot c\cdot \theta_k^2}\right) \\
\le\ & \exp\left(-\frac{(c_1-4c)^2}{72c}\right).
\end{aligned}
$$

□

Finally we are going to use Lemma 8.5.1 with $c_1 = 1/3$. We define V as indicator function of the event $\sum_{i=1}^{n} \frac{w_i}{W} \ln \frac{p_i(S,\pi)}{p(S,\pi)} \ge -c_1$, i.e.

1. $V = 1$ if $\sum_{i=1}^{n} \frac{w_i}{W} \ln \frac{p_i(S,\pi)}{p(S,\pi)} \ge -c_1$.

2. $V = 0$, otherwise.

The probability that A identify the top item can be written as

$$
\begin{aligned}
& \Pr_{(\pi,S)\sim p}[A(S) = \{\pi_1, ..., \pi_k\}] \\
= & \Pr_{(\pi,S)\sim p}[(A(S) = \{\pi_1, ..., \pi_k\}) \wedge (V = 0)] + \\
& \Pr_{(\pi,S)\sim p}[(A(S) = \pi_1) \wedge (V = 1)] \\
\leq & \Pr_{(\pi,S)\sim p}[V = 0] + \sum_{(\pi,S):A(S)=\{\pi_1,...,\pi_k\},V=1} p(S,\pi) \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + \sum_{(\pi,S):A(S)=\{\pi_1,...,\pi_k\},V=1} \\
& \left(e^{c_1} \prod_{i=1}^{n} p_i(S,\pi)^{\frac{w_i}{W}}\right) \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + \sum_{(\pi,S):A(S)=\{\pi_1,...,\pi_k\},V=1} \\
& \left(e^{c_1} \sum_{i=1}^{n} \frac{w_i}{W} \cdot p_i(S,\pi)\right) \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + e^{c_1} \sum_{i=1}^{n} \\
& \frac{w_i}{W} \Pr_{(\pi,S)\sim p_i}[(A(S) = \{\pi_1, ..., \pi_k\}) \wedge (V = 1)] \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + e^{c_1} \sum_{i=1}^{n} \\
& \frac{w_i}{W} \Pr_{(\pi,S)\sim p_i}[A(S) = \{\pi_1, ..., \pi_k\}] \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + e^{c_1} \sum_{i=1}^{n} \frac{w_i}{W} \cdot \frac{1}{2} \\
\leq & \exp\left(-\frac{(c_1 - 4c)^2}{72c}\right) + \frac{e^{c_1}}{2} \\
\leq & \exp\left(-\frac{(1/3 - 4c)^2}{72c}\right) + \frac{3}{4} \\
\leq & \frac{1}{8} + \frac{3}{4} = \frac{7}{8}.
\end{aligned}
$$

The last step comes from the fact that $c < 10^{-4}$. □

The following theorem is very similar to Theorem 8.3.1. For some technical reason, it's not very easy to merge the two proofs. But many parts of proofs of these two theorems are very similar.

Theorem 8.5.2 (Restatement of Theorem 8.3.2)**.** *Assume* $\theta_k > \theta_{k+1}$ *and* $c < 4 \cdot 10^{-4}$*. For any algorithm A (can be adaptive), if A uses* $c \sum_{i:i\le k, \theta_i \le 2\theta_{k+1}} \frac{\theta_{k+1}^2}{(\theta_{k+1}-\theta_i)^2}$ *comparisons of any size (can be l-wise comparison for* $2 \le l \le n$*), then A will identify the top-k items with probability at most* $7/8$*.*

Proof. For notation convenience, we set $w_i = \frac{\theta_{k+1}^2}{(\theta_{k+1}-\theta_i)^2}$ for i such that $i \le k$ and $\theta_i \le 2\theta_{k+1}$. For other i, we set $w_i = 0$. We also set $W = \sum_{i=1}^n w_i$. Then we have $T = cW$.

First of all, we can assume A is deterministic. This is because if A is randomized, we can fix the randomness string which makes A achieves the highest successful probability.

Let $S = (S_1, ..., S_T)$ be the history of algorithm. Each S_t is the comparison result of round t. Notice that since A is deterministic, with $S_1, ..., S_t$, we can determine the labels of items A want to compare in round $t+1$ even when A is adaptive. So there is no point to put the labels of compared items in the history. So we only put the comparison result in the history, i.e S_t is a number in $[n]$ and S is a length-T string of numbers in $[n]$.

Again since A is deterministic, the label A outputs is just a deterministic function of S, we use $A(S)$ to denote it. A outputs correctly if A outputs the label of the top-k items, i.e. $A(S) = \{\pi_1, ..., \pi_k\}$.

We use $p(S, \pi)$ to denote the probability that the items are labeled as π and A has history S. Now consider the case when we set θ_i equals to θ_{k+1} for $i \le k$. In this case the probability of $A(S) = \{\pi_1, ..., \pi_k\}$ should be at most $1/2$ as item $k+1$ and item i have the same weight. We use $p_i(S, \pi)$ to denote the probability that the items are labeled as π and A has history S when θ_i is changed to θ_{k+1}.

Now we prove the following lemma that gives the connection between $p(S, \pi)$ and $p_i(S, \pi)$.

Lemma 8.5.2. *Consider p as a distribution over (π, S). For all $c_1 > 0$, we have*

$$\Pr_{(\pi,S)\sim p}\left[\left(\sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S,\pi)}{p(S,\pi)}\right) \leq -c_1\right] \leq \exp\left(-\frac{(c_1-c)^2}{18c}\right).$$

Proof. Define random variable Z_t to be the following for $t = 1, ..., T$ when (π, S) is sampled from distribution p:

$$Z_t = \sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S_1...S_t,\pi)}{p(S_1...S_t,\pi)}.$$

We have

$$Z_T = \sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S,\pi)}{p(S,\pi)}.$$

Now we want to show that sequence $0, Z_1 + \frac{1}{W}, ..., Z_t + \frac{t}{W}, ..., Z_T + \frac{T}{W}$ forms a supermartigale.

Suppose in round t, given $S_1, ..., S_{t-1}$ and π, Algorithm A compares items in set U_t. Let $\theta_{-i} = \sum_{j\in U_t, j\neq i}\theta_i$. Then we have, with probability $\theta_i/(\theta_i + \theta_{-i})$,

$$Z_t - Z_{t-1} = \frac{w_i}{W}\ln\left(1 - \frac{(\theta_i - \theta_{k+1})\theta_{-i}}{(\theta_{k+1} + \theta_{-i})\theta_i}\right) + \sum_{j\in U_t, j\neq i}\frac{w_j}{W}\ln\left(1 + \frac{\theta_j - \theta_{k+1}}{\theta_{k+1} + \theta_{-j}}\right)$$

Here are two simple facts about ln. For $0 \leq x \leq 1$, $\ln(1+x) \geq x - x^2$. For $0 \leq x \leq 1/2$, $\ln(1-x) \geq -x - x^2$. It's easy to check that for i such that $w_i > 0$, we have $\frac{(\theta_i - \theta_{k+1})\theta_{-i}}{(\theta_{k+1}+\theta_{-i})\theta_i} \leq 1/2$

and $\frac{\theta_i-\theta_{k+1}}{\theta_{k+1}+\theta_{-i}} \le 1$. Therefore, by these two facts, for i such that $w_i > 0$, we have

$$\begin{aligned}
& \frac{\theta_i}{\theta_i+\theta_{-i}} w_i \ln\left(1-\frac{(\theta_i-\theta_{k+1})\theta_{-i}}{(\theta_{k+1}+\theta_{-i})\theta_i}\right) \\
& +\frac{\theta_{-i}}{\theta_i+\theta_{-i}} w_i \ln\left(1+\frac{\theta_i-\theta_{k+1}}{\theta_{k+1}+\theta_{-i}}\right) \\
\ge\ & \frac{w_i}{\theta_i+\theta_{-i}}\left(-\frac{(\theta_i-\theta_{k+1})\theta_{-i}}{\theta_{k+1}+\theta_{-i}}-\frac{(\theta_i-\theta_{k+1})^2\theta_{-i}^2}{(\theta_{k+1}+\theta_{-i})^2\theta_i}+\right. \\
& \left.\frac{(\theta_i-\theta_{k+1})\theta_{-i}}{\theta_{k+1}+\theta_{-i}}-\frac{(\theta_i-\theta_{k+1})^2\theta_{-i}}{(\theta_{k+1}+\theta_{-i})^2}\right) \\
=\ & -w_i\frac{(\theta_{k+1}-\theta_i)^2\theta_{-i}}{(\theta_{k+1}+\theta_{-i})^2\theta_i} = -\frac{\theta_{-i}\theta_{k+1}^2}{(\theta_{k+1}+\theta_{-i})^2\theta_i} \\
\ge\ & -\frac{\theta_{k+1}^2}{(\theta_{k+1}+\theta_{-i})\theta_i} \ge -\frac{\theta_i}{\theta_i+\theta_{-i}}.
\end{aligned}$$

The last step comes from the fact that

$$\theta_{k+1}^2(\theta_i+\theta_{-i}) \le \theta_i^2(\theta_{k+1}+\theta_{-i}).$$

Therefore we have for all t and $S_1, ..., S_{t-1}$,

$$\mathrm{E}[Z_t - Z_{t-1}|S_1, ..., S_{t-1}] \ge -\sum_{i\in U_t}\frac{\theta_i}{W(\theta_i+\theta_{-i})} \ge -\frac{1}{W}.$$

As $Z_1, ..., Z_{t-1}$ can be determined by $S_1, ..., S_{t-1}$, we have for all t and $Z_1, ..., Z_{t-1}$,

$$\begin{aligned}
& \mathrm{E}\left[\left(Z_t+\frac{t}{W}\right)-\left(Z_{t-1}+\frac{t-1}{W}\right)|Z_1-\frac{1}{W}, ..., \right. \\
& \left. Z_{t-1}-\frac{t-1}{W}\right] \ge 0.
\end{aligned}$$

Therefore sequence $0, Z_1+\frac{1}{W}, ..., Z_t+\frac{t}{W}, ..., Z_T+\frac{T}{W}$ forms a supermartingale.

Now we want to bound $|Z_t - Z_{t-1}|$. We know that for $0 \leq x \leq 1$, $|\ln(1+x)| \leq x$ and for $0 \leq x \leq 1/2$, $|\ln(1-x)| \leq 2|x|$. Therefore for i such that $w_i > 0$,

$$|\frac{w_i}{W} \ln\left(1 - \frac{(\theta_i - \theta_{k+1})\theta_{-i}}{(\theta_{k+1} + \theta_{-i})\theta_i}\right)| \leq \frac{w_i}{W} \cdot \frac{2(\theta_i - \theta_{k+1})\theta_{-i}}{(\theta_{k+1} + \theta_{-i})\theta_i}$$
$$\leq \frac{2\theta_{k+1}^2\theta_{-i}}{W\theta_i(\theta_i - \theta_{k+1})(\theta_{k+1} + \theta_{-i})} \leq \frac{2\theta_{k+1}}{W(\theta_k - \theta_{k+1})}$$

and

$$|\frac{w_i}{W} \ln\left(1 + \frac{\theta_i - \theta_{k+1}}{\theta_{k+1} + \theta_{-i}}\right)| \leq \frac{w_i}{W} \cdot \frac{\theta_i - \theta_{k+1}}{\theta_{k+1} + \theta_{-i}} =$$
$$\frac{\theta_{k+1}^2}{W(\theta_i - \theta_{k+1})(\theta_{k+1} + \theta_{-i})} \leq \frac{\theta_{k+1}}{W(\theta_k - \theta_{k+1})} \cdot \frac{\theta_i}{\theta_i + \theta_{-i}}.$$

Therefore, we get

$$|Z_t - Z_{t-1}| \leq \frac{2\theta_{k+1}}{W(\theta_k - \theta_{k+1})} + \sum_{i \in U_t} \frac{\theta_{k+1}}{W(\theta_k - \theta_{k+1})} \cdot$$
$$\frac{\theta_i}{\theta_i + \theta_{-i}} \leq \frac{3\theta_{k+1}}{W(\theta_k - \theta_{k+1})}.$$

Also notice that

$$\left(\frac{\theta_{k+1}}{(\theta_k - \theta_{k+1})}\right)^2 \leq w_k \leq W.$$

Now by Azuma's inequality, we have

$$\begin{aligned}
&\Pr_{(\pi,S)\sim p}[Z_T \leq -c_1] \\
\leq\ & \exp\left(-\frac{(c_1 - \frac{T}{W})^2}{2T(\frac{3\theta_{k+1}}{W(\theta_k - \theta_{k+1})})^2}\right) \\
=\ & \exp\left(-\frac{(c_1 - c)^2(\theta_k - \theta_{k+1})^2 W}{18 \cdot c \cdot \theta_{k+1}^2}\right) \\
\leq\ & \exp\left(-\frac{(c_1 - c)^2}{18c}\right).
\end{aligned}$$

□

After we prove Lemma 8.5.2, the rest of the proof is very similar to Theorem 8.3.1. We omit the argument. □

8.5.2 Lower bounds for arbitrary weights

Again, the following theorem is very similar to Theorem 8.3.1.

Theorem 8.5.3 (Restatement of Theorem 8.3.3)**.** *Assume $c < 1/18$. For any algorithm A (can be adaptive), if A uses $c\sum_{i:i>k}\frac{\theta_i}{\theta_k}$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $7/8$.*

Proof. For notation convenience, we set $w_i = \frac{\theta_i}{\theta_k}$ for $i > k$. For $i \le k$, we set $w_i = 0$. We also set $W = \sum_{i=1}^n w_i$. Then we have $T = cW$.

First of all, we can assume A is deterministic. This is because if A is randomized, we can fix the randomness string which makes A achieves the highest successful probability.

Let $S = (S_1, ..., S_T)$ be the history of algorithm. Each S_t is the comparison result of round t. Notice that since A is deterministic, with $S_1, ..., S_t$, we can determine the labels of items A want to compare in round $t+1$ even when A is adaptive. So there is no point to put the labels of compared items in the history. So we only put the comparison result in the history, i.e S_t is a number in $[n]$ and S is a length-T string of numbers in $[n]$.

Again since A is deterministic, the label A outputs is just a deterministic function of S, we use $A(S)$ to denote it. A outputs correctly if A outputs the label of the top-k items, i.e. $A(S) = \{\pi_1, ..., \pi_k\}$.

We use $p(S,\pi)$ to denote the probability that the items are labeled as π and A has history S. Now consider the case when we set θ_i equals to θ_k for $i > k$. In this case the probability of $A(S) = \{\pi_1, ..., \pi_k\}$ should be at most $1/2$ as item k and item i have the same weight. We use $p_i(S,\pi)$ to denote the probability that the items are labeled as π and A has history S when θ_i is changed to θ_k.

Now we prove the following lemma that gives the connection between $p(S,\pi)$ and $p_i(S,\pi)$.

Lemma 8.5.3. *Consider p as a distribution over (π, S). For all $c_1 > 0$, we have*

$$\begin{aligned} &\Pr_{(\pi,S)\sim p}\left[\left(\sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S,\pi)}{p(S,\pi)}\right)\leq -c_1\right] \\ &\leq \exp\left(-\frac{(c_1/c-1)^2T}{8}\right). \end{aligned}$$

Proof. Define random variable Z_t to be the following for $t = 1, ..., T$ when (π, S) is sampled from distribution p:

$$Z_t = \sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S_1...S_t,\pi)}{p(S_1...S_t,\pi)}.$$

We have

$$Z_T = \sum_{i=1}^{n}\frac{w_i}{W}\ln\frac{p_i(S,\pi)}{p(S,\pi)}.$$

Now we want to show that sequence $0, Z_1+\frac{1}{W}, ..., Z_t+\frac{t}{W}, ..., Z_T+\frac{T}{W}$ forms a supermartingale.

Suppose in round t, given $S_1, ..., S_{t-1}$ and π, Algorithm A compares items in set U_t. Let $\theta_{-i} = \sum_{j\in U_t, j\neq i}\theta_i$. Then we have, with probability $\theta_i/(\theta_i+\theta_{-i})$,

$$\begin{aligned} & Z_t - Z_{t-1} \\ = \; & \frac{w_i}{W}\ln\left(1+\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_k+\theta_{-i})\theta_i}\right) \\ & + \sum_{j\in U_t, j\neq i}\frac{w_j}{W}\ln\left(1-\frac{\theta_k-\theta_j}{\theta_k+\theta_{-j}}\right) \\ = \; & -\frac{w_i}{W}\ln\left(1-\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_i+\theta_{-i})\theta_k}\right) \\ & - \sum_{j\in U_t, j\neq i}\frac{w_j}{W}\ln\left(1+\frac{\theta_k-\theta_j}{\theta_j+\theta_{-j}}\right) \end{aligned}$$

We are going to use a simple fact about ln: for all $x > -1$, $\ln(1+x) \le x$.

$$\begin{aligned} & -\left(\frac{\theta_i}{\theta_i+\theta_{-i}} w_i \ln\left(1-\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_i+\theta_{-i})\theta_k}\right)\right. \\ & \left.+\frac{\theta_{-i}}{\theta_i+\theta_{-i}} w_i \ln\left(1+\frac{\theta_k-\theta_i}{\theta_i+\theta_{-i}}\right)\right) \\ \ge\ & -\frac{w_i}{\theta_i+\theta_{-i}}\left(-\frac{(\theta_k-\theta_i)\theta_{-i}\theta_i}{(\theta_i+\theta_{-i})\theta_k}+\frac{(\theta_k-\theta_i)\theta_{-i}}{\theta_i+\theta_{-i}}\right) \\ =\ & -\frac{(\theta_k-\theta_i)^2\theta_{-i}\theta_i}{(\theta_i+\theta_{-i})^2\theta_k^2} \\ \ge\ & -\frac{\theta_i}{\theta_i+\theta_{-i}}. \end{aligned}$$

Therefore we have for all t and $S_1, ..., S_{t-1}$,

$$\mathrm{E}[Z_t - Z_{t-1} | S_1, ..., S_{t-1}] \ge -\sum_{i\in U_t} \frac{\theta_i}{W(\theta_i+\theta_{-i})} \ge -\frac{1}{W}.$$

As $Z_1, ..., Z_{t-1}$ can be determined by $S_1, ..., S_{t-1}$, we have for all t and $Z_1, ..., Z_{t-1}$,

$$\begin{aligned} & \mathrm{E}\left[\left(Z_t+\frac{t}{W}\right)-\left(Z_{t-1}+\frac{t-1}{W}\right)|Z_1-\frac{1}{W}, ...,\right. \\ & \left. Z_{t-1}-\frac{t-1}{W}\right] \ge 0. \end{aligned}$$

Therefore sequence $0, Z_1+\frac{1}{W}, ..., Z_t+\frac{t}{W}, ..., Z_T+\frac{T}{W}$ forms a supermartingale.

Now we want to bound $|Z_t - Z_{t-1}|$. We know that for $0 \le x$, $|\ln(1+x)| \le x$. Therefore for i such that $w_i > 0$,

$$\begin{aligned} & |\frac{w_i}{W}\ln\left(1+\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_k+\theta_{-i})\theta_i}\right)| \le \frac{w_i}{W}\cdot\frac{(\theta_k-\theta_i)\theta_{-i}}{(\theta_k+\theta_{-i})\theta_i} \\ & \le \frac{(\theta_k-\theta_i)\theta_{-i}}{W(\theta_k+\theta_{-i})\theta_k} \le \frac{1}{W} \end{aligned}$$

and

$$
\begin{aligned}
&|\frac{w_i}{W}\ln\left(1+\frac{\theta_k-\theta_i}{\theta_i+\theta_{-i}}\right)| \le \frac{w_i(\theta_k-\theta_i)}{W(\theta_i+\theta_{-i})} \\
&= \frac{\theta_i(\theta_k-\theta_i)}{W\theta_k(\theta_i+\theta_{-i})} \le \frac{1}{W}\cdot\frac{\theta_i}{\theta_i+\theta_{-i}}.
\end{aligned}
$$

Therefore, we get

$$
|Z_t - Z_{t-1}| \le \frac{1}{W} + \sum_{i\in U_t}\frac{1}{W}\cdot\frac{\theta_i}{\theta_i+\theta_{-i}} \le \frac{2}{W}.
$$

Now by Azuma's inequality, we have

$$
\begin{aligned}
\Pr_{(\pi,S)\sim p}[Z_T \le -c_1] \quad &\le \quad \exp\left(-\frac{(c_1-\frac{T}{W})^2}{2T(\frac{2}{W})^2}\right) \\
&= \quad \exp\left(-\frac{(c_1/c-1)^2T}{8}\right).
\end{aligned}
$$

□

After we prove Lemma 8.5.3, the rest of the proof is very similar to Theorem 8.3.1 by picking $c_1 = 1/3$. We omit the argument. □

Theorem 8.5.4 (Restatement of Theorem 8.3.4)**.** *For any algorithm A (can be adaptive), if A uses $k/4$ comparisons of any size (can be l-wise comparison for $2 \le l \le n$), then A will identify the top-k items with probability at most $2/3$.*

Proof. First of all, we can assume A is deterministic. This is because if A is randomized, we can fix the randomness string which makes A achieves the highest successful probability.

Let $S = (S_1, ..., S_T)$ be the history of algorithm. Each S_t is the comparison result of round t. Notice that since A is deterministic, with $S_1, ..., S_t$, we can determine the labels of items A want to compare in round $t+1$ even when A is adaptive. So there is no point to put the labels of compared items in the history. So we only put the comparison result in the history, i.e S_t is a number in $[n]$ and S is a length-T string of numbers in $[n]$.

Again since A is deterministic, the label A outputs is just a deterministic function of S, we use $A(S)$ to denote it. A outputs correctly if A outputs the label of the top-k items, i.e. $A(S) = \{\pi_1, ..., \pi_k\}$.

We use $p(S, \pi)$ to denote the probability that the items are labeled as π and A has history S. Now consider the case when we set θ_i equals to θ_{k+1} for $i \leq k$. In this case the probability of $A(S) = \{\pi_1, ..., \pi_k\}$ should be at most $1/2$ as item $k+1$ and item i have the same weight. We use $p_i(S, \pi)$ to denote the probability that the items are labeled as π and A has history S when θ_i is changed to θ_{k+1}.

We define $N(\pi, S)$ as the set of items among top-k items such that they are not chosen as the favorite items by algorithm A in history S with labels π. As there are only $k/4$ comparisons, $N(\pi, S) \leq 3/4$ for all π, S.

Now we prove the following simple lemma that gives the connection between $p(S, \pi)$ and $p_i(S, \pi)$ for all $i \in N(\pi, S)$.

Lemma 8.5.4.

$$\forall \pi, S, i \in N(\pi, S), p_i(S, \pi) \geq p(S, \pi).$$

Proof. We write $p(S, \pi)$ as

$$p(S, \pi) = \prod_{t=1}^{T} p(S_t, \pi | S_1 ... S_{t-1}).$$

And similarly $p_i(S, \pi)$ as

$$p_i(S, \pi) = \prod_{t=1}^{T} p_i(S_t, \pi | S_1 ... S_{t-1}).$$

Consider the comparison in round t given $S_1, S_2, ..., S_t, \pi$. There are two cases

1. i-th item is not compared in round t: The change of θ_i does not change $p(S_t, \pi | S_1 ... S_{t-1})$. So $p(S_t, \pi | S_1 ... S_{t-1}) = p_i(S_t, \pi | S_1 ... S_{t-1})$.

2. i-th item is compared in round t: We know the i-th item is not the favorite item of round t in this history. Therefore decreasing θ_i to θ_{k+1} will increase $p(S_t, \pi | S_1 ... S_{t-1})$. So $p(S_t, \pi | S_1 ... S_{t-1}) \leq p_i(S_t, \pi | S_1 ... S_{t-1})$.

Thus we always have $p(S_t, \pi | S_1 ... S_{t-1}) \leq p_i(S_t, \pi | S_1 ... S_{t-1})$. By multiplying things together we get the statement of this lemma. □

Finally we have

$$
\begin{aligned}
& \Pr_{(\pi,S)\sim p}[A(S) = \{\pi_1, ..., \pi_k\}] \\
= & \sum_{\pi,S,A(S)=\{\pi_1,...,\pi_k\}} p(\pi, S) \\
\leq & \sum_{\pi,S,A(S)=\{\pi_1,...,\pi_k\}} \frac{1}{|N(\pi, S)|} \sum_{i \in N(\pi,S)} p_i(\pi, S) \\
\leq & \sum_{\pi,S,A(S)=\{\pi_1,...,\pi_k\}} \frac{1}{|N(\pi, S)|} \sum_{i \in \{1,...,k\}} p_i(\pi, S) \\
\leq & \sum_{\pi,S,A(S)=\{\pi_1,...,\pi_k\}} \frac{4}{3k} \sum_{i \in \{1,...,k\}} p_i(\pi, S) \\
= & \frac{4}{3k} \sum_{i \in \{1,...,k\}} \sum_{\pi,S,A(S)=\{\pi_1,...,\pi_k\}} p_i(\pi, S) \\
\leq & \frac{4}{3k} \sum_{i \in \{1,...,k\}} \frac{1}{2} \\
= & \frac{2}{3}.
\end{aligned}
$$

□

Theorem 8.5.5 (Restatement of Theorem 8.3.5)**.** *Assume $c < 1/2$. For any algorithm A (can be adaptive), if A uses $\frac{cn}{l}$ comparisons of size at most l (can be 2-wise, 3-wise,...,l-wise comparisons), then A will identify the top-k items with probability at most $7/8$.*

Proof. We are going to prove by contradiction. Suppose there's some A uses $\frac{cn}{l}$ comparisons of size at most l and identify the top-k items with probability more than $7/8$.

Now consider another task where the goal is just to make sure k-th item appeared in some comparison or $(k+1)$-th item appeared in some comparison. Notice that when some algorithm fails this new task, then the algorithm cannot output top-k items with probability better than $1/2$ because when both k-th item and $(k+1)$-th item are not compared, the algorithm should output them with same probability for identifying top-k items. So algorithm A should solve the new task with probability more than $3/4$.

For the new task, it's easy to see that the best strategy is to always use l-wise comparison and compare $\frac{cn}{l} \cdot l$ different items. The probability of having either k-th item or $(k+1)$-th item compared is

$$1 - \left(1 - \frac{2}{n}\right)^{cn} \leq 1 - \frac{1}{4^{2c}} \leq 3/4.$$

Here we need to use the fact that $n \geq 4$ (when $n < 4$, the statement of the theorem is trivial). Now we get a contradiction. $\square$

Bibliography

[1] N. Ailon. Active learning ranking from pairwise preferences with almost optimal query complexity. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2011.

[2] N Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008.

[3] M Ajtai, J Komlos, W L Steiger, and E Szemeredi. Deterministic selection in o(loglog n) parallel time. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, pages 188–195, New York, NY, USA, 1986. ACM.

[4] Miklós Ajtai, János Komlós, and Endre Szemerédi. An o(n log n) sorting network. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 1–9, 1983.

[5] Noga Alon and Yossi Azar. The average complexity of deterministic and randomized parallel comparison-sorting algorithms. *SIAM J. Comput.*, 17(6):1178–1192, 1988.

[6] Noga Alon and Yossi Azar. Sorting, approximate sorting, and searching in rounds. *SIAM J. Discrete Math.*, 1(3):269–280, 1988.

[7] Noga Alon, Yossi Azar, and Uzi Vishkin. Tight complexity bounds for parallel comparison sorting. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 502–510, 1986.

[8] Noga Alon, Noam Nisan, Ran Raz, and Omri Weinstein. Welfare maximization with limited interaction. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1499–1512, 2015.

[9] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Learning prices for repeated auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 1169–1177, 2013.

[10] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 622–630, 2014.

[11] Masaki Aoyagi. Bid rotation and collusion in repeated auctions. *Journal of Economic Theory*, 112(1):79–105, 2003.

[12] Masaki Aoyagi. Efficient collusion in repeated auctions with communication. *Journal of Economic Theory*, 134(1):61–92, 2007.

[13] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1503–1510, New York, NY, USA, 2012. ACM.

[14] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[15] Itai Ashlagi, Constantinos Daskalakis, and Nima Haghpanah. Sequential mechanisms with ex-post participation guarantees. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 213–214, 2016.

[16] Sepehr Assadi. Combinatorial auctions do need modest interaction. *manuscript*, 2017.

[17] Susan Athey and Kyle Bagwell. Optimal collusion with private information. *RAND Journal of Economics*, 32(3):428–65, 2001.

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

[19] László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, 2003.

[20] Moshe Babaioff, Robert D. Kleinberg, and Aleksandrs Slivkins. Truthful mechanisms with implicit payment computation. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 43–52, New York, NY, USA, 2010. ACM.

[21] Moshe Babaioff, Brendan Lucier, Noam Nisan, and Renato Paes Leme. On the efficiency of the walrasian mechanism. In *the 16th Annual ACM Conference on Economics and Computation (EC)*, 2014.

[22] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. Characterizing truthful multi-armed bandit mechanisms: Extended abstract. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pages 79–88, New York, NY, USA, 2009. ACM.

[23] Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1025–1035, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics.

[24] T. P. Ballinger and N. T. Wilcox. Decisions, error and heterogeneity. *The Economic Journal*, 107(443):1090–1105, 1997.

[25] Xiaohui Bei and Zhiyi Huang. Bayesian Incentive Compatibility via Fractional Assignments. In *the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[26] Dirk Bergemann and Juuso Vlimki. Learning and strategic pricing. *Econometrica*, 64(5):1125–49, 1996.

[27] Kshipra Bhawalkar and Tim Roughgarden. Welfare guarantees for combinatorial auctions with item bidding. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 700–709, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics.

[28] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007.

[29] Manuel Blum, Robert W. Floyd, Vaughan Ronald Pratt, Ronald L. Rivest, and Robert Endre Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.

[30] Béla Bollobás and Graham Brightwell. Parallel selection with high probability. *SIAM J. Discrete Math.*, 3(1):21–31, 1990.

[31] Béla Bollobás and Pavol Hell. Sorting and graphs. In Ivan Rival, editor, *Graphs and Order*, volume 147 of *NATO ASI Series*, pages 169–184. Springer Netherlands, 1985.

[32] Béla Bollobás and Andrew Thomason. Parallel sorting. *Discrete Applied Mathematics*, 6(1):1 – 11, 1983.

[33] R. Bradley and M. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[34] M. Braverman, J. Mao, and M. S. Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proccedings of the Annual Symposium on the Theory of Computing (STOC)*, 2016.

[35] M. Braverman and E. Mossel. Noisy sorting without resampling. In *Proceedings of the ACM-SIAM symposium on discrete algorithms (SODA)*, 2008.

[36] Mark Braverman, Jieming Mao, Jon Schneider, and S. Matthew Weinberg. Multi-armed bandit problems with strategic arms. *In Submission, https://arxiv.org/abs/1706.09060*, 2017.

[37] Mark Braverman, Jieming Mao, Jon Schneider, and S. Matthew Weinberg. Selling to a no-regret buyer. *EC*, 2018.

[38] Mark Braverman, Jieming Mao, and S. Matthew Weinberg. Interpolating between truthful and non-truthful mechanisms for combinatorial auctions. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1444–1457, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[39] Mark Braverman, Jieming Mao, and S. Matthew Weinberg. On simultaneous two-player combinatorial auctions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018.

[40] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 268–276, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[41] Mark Braverman and Elchanan Mossel. Sorting from noisy information. *CoRR*, abs/0910.1191, 2009.

[42] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[43] Patrick Briest, Piotr Krysta, and Berthold Vöcking. Approximation techniques for utilitarian mechanism design. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 39–48, New York, NY, USA, 2005. ACM.

[44] Sebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

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

[46] Róbert Busa-Fekete, Balázs Szörényi, Weiwei Cheng, Paul Weng, and Eyke Hüllermeier. Top-k selection based on adaptive sampling of noisy preferences. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1094–1102, 2013.

[47] Yang Cai and Constantinos Daskalakis. Learning multi-item auctions with (or without) samples. In *FOCS*, 2017.

[48] Yang Cai and Christos H. Papadimitriou. Simultaneous bayesian auctions and computational complexity. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, pages 895–910, 2014.

[49] E. J. Candés. Modern statistical estimation via oracle inequalities. *Acta Numerica*, 15:257–325, 2006.

[50] Andrew Caplin and Barry Nalebuff. Aggregation and social choice: A mean voter theorem. *Econometrica*, 59(1):1–23, 1991.

[51] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 94–99, 1983.

[52] Sylvain Chassang. Calibrated incentive contracts. *Econometrica*, 81(5):1935–1971, 2013.

[53] Sabyasachi Chatterjee, Adityanand Guntuboyina, and Bodhisattva Sen. On risk bounds in isotonic and other shape restricted regression problems. 43(4):1774–1800, 2014.

[54] Sabyasachi Chatterjee, Adityanand Guntuboyina, and Bodhisattva Sen. On matrix estimation under monotonicity constraints. arXiv preprint arXiv:1506.03430, 2015.

[55] J. Chen, X. Chen, Q. Zhang, and Y. Zhou. Adaptive multiple-arm identification. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[56] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013.

[57] X. Chen, S. Gopi, J. Mao, and J. Schneider. Competitive analysis of the top-k ranking problem. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017.

[58] Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 193–202, New York, NY, USA, 2013. ACM.

[59] Xi Chen, Yuanzhi Li, and Jieming Mao. A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018.

[60] Y. Chen and C. Suh. Spectral MLE: Top-K rank aggregation from pairwise comparisons. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[61] Yuxin Chen and Changho Suh. Spectral MLE: top-k rank aggregation from pairwise comparisons. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 371–380, 2015.

[62] George Christodoulou, Annamária Kovács, and Michael Schapira. Bayesian combinatorial auctions. In *ICALP*, 2008.

[63] Edward H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11(1):17–33, 1971.

[64] Richard Cole. Parallel merge sort. *SIAM J. Comput.*, 17(4):770–785, 1988.

[65] Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 243–252, New York, NY, USA, 2014. ACM.

[66] I. Csiszar and J. Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[67] Amit Daniely, Michael Schapira, and Gal Shahaf. Inapproximability of truthful mechanisms via generalizations of the vc dimension. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 401–408, New York, NY, USA, 2015. ACM.

[68] Constantinos Daskalakis and Vasilis Syrgkanis. Learning in auctions: Regret is hard, envy is easy. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 219–228, 2016.

[69] D. Davidson and J. Marschak. Experimental tests of a stochastic decision theory. *Measurement: Definitions and theories*, pages 233–269, 1959.

[70] Nikhil Devanur, Jamie Morgenstern, Vasilis Syrgkanis, and S. Matthew Weinberg. Simple auctions with simple strategies. *Manuscript*, 2015. http://www.cs.cmu.edu/ jamiemmt/papers/draft.pdf.

[71] Nikhil R. Devanur, Zhiyi Huang, and Christos-Alexandros Psomas. The sample complexity of auctions with side information. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 426–439, New York, NY, USA, 2016. ACM.

[72] Nikhil R. Devanur and Sham M. Kakade. The price of truthfulness for pay-per-click auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pages 99–106, New York, NY, USA, 2009. ACM.

[73] Nikhil R. Devanur, Yuval Peres, and Balasubramanian Sivan. Perfect bayesian equilibria in repeated sales. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 983–1002, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics.

[74] Shahar Dobzinski. Two randomized mechanisms for combinatorial auctions. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 89–103, 2007.

[75] Shahar Dobzinski. An impossibility result for truthful combinatorial auctions with submodular valuations. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 139–148, 2011.

[76] Shahar Dobzinski. Breaking the logarithmic barrier for truthful combinatorial auctions with submodular bidders. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 940–948, New York, NY, USA, 2016. ACM.

[77] Shahar Dobzinski. Computational efficiency requires simple taxation. In *FOCS 2016*, 2016.

[78] Shahar Dobzinski, Hu Fu, and Robert Kleinberg. On the complexity of computing an equilibrium in combinatorial auctions. In *the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

[79] Shahar Dobzinski, Noam Nisan, and Sigal Oren. Economic efficiency requires interaction. In *the 46th annual ACM symposium on Theory of computing (STOC)*, 2014.

[80] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 610–618. ACM, 2005.

[81] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 644–652. ACM, 2006.

[82] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, pages 1064–1073, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics.

[83] Shahar Dobzinski and Jan Vondrák. The computational complexity of truthfulness in combinatorial auctions. In *ACM Conference on Electronic Commerce, EC '12, Valencia, Spain, June 4-8, 2012*, pages 405–422, 2012.

[84] Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1107–1116, 2012.

[85] Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E. Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient learning and auction design. In *FOCS*, 2017.

[86] Shaddin Dughmi, Tim Roughgarden, and Qiqi Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 149–158, 2011.

[87] Shaddin Dughmi and Jan Vondrák. Limitations of randomized mechanisms for combinatorial auctions. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 502–511, 2011.

[88] Pavol Duris, Zvi Galil, and Georg Schnitger. Lower bounds on communication complexity. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 81–91, New York, NY, USA, 1984. ACM.

[89] Paul Dütting, Felix Fischer, and David C. Parkes. Simplicity-expressiveness tradeoffs in mechanism design. In *the 13th Annual ACM Conference on Economics and Computation (EC)*, 2011.

[90] Paul Dütting, Felix Fischer, and David C. Parkes. Expressiveness and robustness of first-price position auctions. In *the 16th Annual ACM Conference on Economics and Computation (EC)*, 2014.

[91] Paul Dütting and Thomas Kesselheim. Algorithms against anarchy: Understanding non-truthful mechanisms. In *the 17th Annual ACM Conference on Economics and Computation (EC)*, 2015.

[92] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International World Wide Web Conference*, 2001.

[93] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA, 2001. ACM.

[94] EB Dynkin. The optimum choice of the instant for stopping a markov process. In *Sov. Math. Dokl*, volume 4, pages 627–629, 1963.

[95] Brian Eriksson. Learning to top-k search using pairwise comparisons. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, pages 265–273, 2013.

[96] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*, 66(4):614–656, 2003.

[97] Uriel Feige. On maximizing welfare when utility functions are subadditive. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 41–50, New York, NY, USA, 2006. ACM.

[98] Uriel Feige and Jan Vondrak. The allocation problem with submodular utility functions. In *In Proc. of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS*, 2006.

[99] Michal Feldman, Ophir Friedler, Jamie Morgenstern, and Guy Reiner. Simple mechanisms for agents with complements. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 251–267, New York, NY, USA, 2016. ACM.

[100] Michal Feldman, Hu Fu, Nick Gravin, and Brendan Lucier. Simultaneous auctions are (almost) efficient. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 201–210. ACM, 2013.

[101] Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial auctions via posted prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 123–135, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics.

[102] P. C. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical psychology*, 10(4):327–352, 1973.

[103] Dean P Foster and Rakesh V Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1):40–55, 1997.

[104] Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 5–22, New York, NY, USA, 2014. ACM.

[105] J.C. Gittins and D.M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani, editor, *Progress in Statistics*, pages 241–266. North-Holland, Amsterdam, 1974.

[106] Gagan Goel, Vahab Mirrokni, and Renato Paes Leme. Polyhedral clinching auctions and the adwords polytope. In *the 44th annual ACM symposium on Theory of computing (STOC)*, 2012.

[107] Gagan Goel, Vahab Mirrokni, and Renato Paes Leme. Clinching auctions with online supply. In *the 23th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013.

[108] Gagan Goel, Vahab Mirrokni, and Renato Paes Leme. Clinching auctions beyond hard budget constraints. In *the 16th Annual ACM Conference on Economics and Computation (EC)*, 2014.

[109] Yannai A. Gonczarowski and Noam Nisan. Efficient empirical revenue maximization in single-parameter auction environments. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 856–868, New York, NY, USA, 2017. ACM.

[110] Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.

[111] A. Guntuboyina, D. Lieu, S. Chatterjee, and B. Sen. Spatial adaptation in trend filtering. arXiv preprint arXiv:1702.05113, 2017.

[112] Adityanand Guntuboyina and Bodhisattva Sen. Global risk bounds and adaptation in univariate convex regression. *Probab. Theory Related Fields*, 2013. *To appear*, available at http://arxiv.org/abs/1305.1648.

[113] Roland Häggkvist and Pavol Hell. Parallel sorting with constant time for comparisons. *SIAM J. Comput.*, 10(3):465–472, 1981.

[114] James Hannan. Approximation to bayes risk in repeated play. In *Contributions to the Theory of Games*, pages 3:97–139, 1957.

[115] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[116] Jason Hartline, Darrell Hoy, and Sam Taggart. Price of anarchy for auction revenue. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, 2014.

[117] Jason D. Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian Incentive Compatibility via Matchings. In *the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[118] Jason D. Hartline and Brendan Lucier. Bayesian Algorithmic Mechanism Design. In *the 42nd ACM Symposium on Theory of Computing (STOC)*, 2010.

[119] 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:1606.08842v2, 2016.

[120] Paul Dütting Monika Henzinger and Martin Starnberger. Valuation compressions in vcg-based combinatorial auctions. In *Workshop on Internet and Network Economics (WINE)*, 2013.

[121] Charles Antony Richard Hoare. Algorithm 65: Find. *Communications of the ACM*, 4(7):321–322, 1961.

[122] Nicole Immorlica, Brendan Lucier, Emmanouil Pountourakis, and Samuel Taggart. Repeated sales with multiple strategic buyers. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 167–168. ACM, 2017.

[123] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *Proceedings of Conference on Learning Theory (COLT)*, 2014.

[124] K. Jamieson and R. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, 2011.

[125] 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, 2013.

[126] Paul Johnson and Jacques Robert. Collusion in a model of repeated auctions. Cahiers de recherche, Universite de Montreal, Departement de sciences economiques, 1999.

[127] Sham M. Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013.

[128] Adam Kalai and Santosh Vempala. Geometric algorithms for online optimization. In *Journal of Computer and System Sciences*, pages 26–40, 2002.

[129] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, October 2005.

[130] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Symposium on Theory of computing (STOC)*, 2007.

[131] Stavros G. Kolliopoulos and Clifford Stein. *Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs*, pages 153–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[132] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole dté de Probabilités de Saint-Flour XXXVIII-2008*, volume 38. Springer Science & Business Media, 2011.

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

[134] Piotr Krysta and Berthold Vöcking. Online mechanism design (randomized rounding on the fly). In *Automata, Languages, and Programming*, pages 636–647. Springer, 2012.

[135] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, New York, NY, USA, 1997.

[136] Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. 2002.

[137] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22, March 1985.

[138] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 817–824. Curran Associates, Inc., 2008.

[139] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005.

[140] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *the 3rd Annual ACM Conference on Electronic Commerce (EC)*, 2001.

[141] Frank Thomson Leighton. Tight bounds on the complexity of parallel sorting. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 71–80, 1984.

[142] Frank Thomson Leighton and Yuan Ma. Tight bounds on the size of fault-tolerant merging and sorting networks with destructive faults. *SIAM Journal on Computing*, 29(1):258–273, 2000.

[143] Siqi Liu and Christos-Alexandros Psomas. On the competition complexity of dynamic mechanism design. *CoRR*, abs/1709.07955, 2017.

[144] T. Lu and C. Boutilier. Learning mallows models with pairwise preferences. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

[145] R. D. Luce. *Individual choice behavior: A theoretical analysis*. New York: Wiley, 1959.

[146] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Sorting noisy data with partial information. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 515–528, New York, NY, USA, 2013. ACM.

[147] Randolph McAfee and John McMillan. Bidding rings. *American Economic Review*, 82(3):579–99, 1992.

[148] John McCarthy. Measures of the value of information. *Proceedings of the National Academy of Sciences*, 42(9):654–655, 1956.

[149] D. McFadden. Conditional logit analysis of qualitative choice behaviour. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press New York, New York, NY, USA, 1973.

[150] D. H. McLaughlin and R. D. Luce. Stochastic transitivity and cancellation of preferences between bitter-sweet solutions. *Psychonomic Science*, 2(1–12):89–90, 1965.

[151] Vahab S. Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Dynamic auctions with bank accounts. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 387–393, 2016.

[152] Vahab S. Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Optimal dynamic mechanisms with ex-post IR via bank accounts. *CoRR*, abs/1605.08840, 2016.

[153] Vahab S. Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*, pages 70–77, 2008.

[154] S. Mohajer and C. Suh. Active top-k ranking from noisy comparisons. In *Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016.

[155] Jamie Morgenstern and Tim Roughgarden. The pseudo-dimension of near-optimal auctions. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 136–144, Cambridge, MA, USA, 2015. MIT Press.

[156] Jamie Morgenstern and Tim Roughgarden. Learning simple auctions. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1298–1318, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[157] Roger B. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[158] Hamid Nazerzadeh, Amin Saberi, and Rakesh Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 179–188, New York, NY, USA, 2008. ACM.

[159] S. Negahban, S. Oh, and D. Sha. Rank Centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017.

[160] Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, pages 1–18, New York, NY, USA, 2015. ACM.

[161] Noam Nisan and Amir Ronen. Algorithmic Mechanism Design (Extended Abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC)*, 1999.

[162] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1):211–219, February 1993.

[163] Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. Sequential auctions and externalities. In *SODA*, 2012.

[164] Christos Papadimitriou, George Pierrakos, Christos-Alexandros Psomas, and Aviad Rubinstein. On the complexity of dynamic mechanism design. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1458–1475, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[165] Christos H. Papadimitriou, Michael Schapira, and Yaron Singer. On the hardness of being truthful. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008.

[166] Christos H. Papadimitriou and Michael Sipser. Communication complexity. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 196–200, New York, NY, USA, 1982. ACM.

[167] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. In *Proceedings of the Sixth International Conference on Educational Data Mining*, EDM '13, 2013.

[168] Prabhakar Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *J. Comput. Syst. Sci.*, 37(2):130–143, October 1988.

[169] A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

[170] Rüdiger Reischuk. A fast probabilistic parallel sorting algorithm. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 212–219, 1981.

[171] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[172] Kevin Roberts. The characterization of implementable choice rules. *Jean-Jacques Laffont, editor, Aggregation and Revelation of Preferences*, page 321349, 1979. Papers presented at the 1st European Summer Workshop of the Econometric Society, North-Holland.

[173] Tim Roughgarden. The price of anarchy in games of incomplete information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 862–879, New York, NY, USA, 2012. ACM.

[174] Tim Roughgarden. Barriers to near-optimal equilibria. In *FOCS*, 2014.

[175] N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 63(2):934–959, 2016.

[176] N. B. Shah, S. Balakrishnan, and M. J. Wainwright. Feeling the bern: Adaptive estimators for bernoulli probabilities of pairwise comparisons. arXiv preprint arXiv:1603.06881v1, 2016.

[177] N. B. Shah and M. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. arXiv preprint arXiv:1512.08949, 2015.

[178] Andrzej Skrzypacz and Hugo Hopenhayn. Tacit collusion in repeated auctions. *Journal of Economic Theory*, 114(1):153–169, 2004.

[179] C. Suh, V. Tan, and R. Zhao. Adversarial top-K ranking. *IEEE Transactions on Information Theory (to appear)*, 2017. DOI 10.1109/TIT.2017.2659660.

[180] Vasilis Syrgkanis and Eva Tardos. Bayesian sequential auctions. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 929–944. ACM, 2012.

[181] Vasilis Syrgkanis and Eva Tardos. Composable and efficient mechanisms. In *STOC*, 2013.

[182] B. Szörényi, R. Busa-Fekete, A. Paul, and E. Hüllermeier. Online rank elicitation for plackett-luce: A dueling bandits approach. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[183] L. L. Thurstone. A law of comparative judgement. *Psychological Reviews*, 34(4):273, 1927.

[184] Kenneth Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2003.

[185] A. Tversky. Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281–299, 1972.

[186] Leslie G. Valiant. Parallelism in comparison problems. *SIAM J. Comput.*, 4(3):348–355, 1975.

[187] William Vickrey. Counterspeculations, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16(1):8–37, 1961.

[188] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 67–74, 2008.

[189] F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[190] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM.

[191] Y. Zhou, X. Chen, and J. Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In *Proceedings of International Conference on Machine Learning (ICML)*, 2014.