SEQUENTIAL DECISION MAKING WITH STRATEGIC AGENTS AND LIMITED FEEDBACK

A Dissertation
Presented to
The Academic Faculty

By

Bhuvesh Kumar

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology

December 2022

SEQUENTIAL DECISION MAKING WITH STRATEGIC AGENTS AND LIMITED FEEDBACK

Thesis committee:

Dr. Jacob Abernethy (Co-advisor)
School of Computer Science
Georgia Institute of Technology

Dr. Jamie Morgenstern (Co-advisor)
Paul G. Allen School of Computer Science & Engineering
University of Washington

Dr. Vidya Muthukumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Florian Schäfer
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Sahil Singla
School of Computer Science
Georgia Institute of Technology

Date approved: December 11, 2022

For my parents

ACKNOWLEDGMENTS

I am most thankful to my advisors, Jacob Abernethy and Jamie Morgenstern, who have been very supportive throughout my journey. I am grateful to have great advisors who were very generous with their time even while being involved with multiple projects and were also very kind to have been very supportive during my personal times of struggles as well.

I would like to thank the members of my thesis committee, Sahil Singla, Vidya Muthukumar, and Florian Schäferfor their help in the preparation of this work. They helped to shed new light on many of my ideas and gave me very valuable feedback about my thesis.

I am thankful to all coauthors I have had the pleasure of working with. I thank all my friends and fellow grad students at Georgia Tech.

I am thankful for my Atlanta family members, Lindsey and Dorene who have always motivated me to be the best version of myself and have given me a home away from home.

Lastly, I would like to of course thank my family, especially my parents for always helping, encouraging, and putting up with any endeavor I have pursued throughout my life, and my brother and sister-in-law who have always been by my side cheering me on.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Sequential decision-making is a natural model for machine learning applications where the learner must make online decisions in real-time and simultaneously learn from the sequential data to make better decisions in the future. Classical work has focused on variants of the problem based on the data distribution being either stochastic or adversarial, or based on the feedback available to the learner's decisions which could be either partial or complete. With the rapid rise of large online markets, sequential learning methods have increasingly been deployed in complex multi-agent systems where agents may behave strategically to optimize for their own personal objectives. This has added a new dimension to the sequential decision-making problem where the learner must account for the strategic behavior of the agents it is learning from who might want to steer its future decisions in their favor.

This thesis aims to design effective online decision-making algorithms from the point of view of both the system designers aiming to learn in environments with strategic agents and limited feedback and also the strategic agents seeking to optimize personal objectives. In Part I of the thesis, we focus on repeated auctions and design mechanisms where the auctioneer can effectively learn in presence of strategic bidders, and conversely, address how agents can bid in repeated auctions or use data-poisoning attacks to maximize their own objectives.

In Part II, we consider an online learning setting where feedback about the learner's decisions is expensive to obtain. We introduce an online learning algorithm inspired by techniques from active learning that can fast forward a small fraction of more informative examples ahead in the queue. This allows the learner to obtain the same performance as the optimal online algorithm but only by querying feedback on a very small fraction of points. Finally, in Part III of the thesis, we consider a new learning objective for stochastic multi-arm bandits that promotes merit-based fairness in opportunity for individuals and groups.

CHAPTER 1
INTRODUCTION AND BACKGROUND

With the rise of internet platforms, we as humans interact with machine learning algorithms every day. Whenever we use a video-watching app, an algorithm recommends us a video; whenever we do online shopping, machine learning algorithms rank the search results; whenever we book a ride using a ride-sharing app, an algorithm decides which driver will come to pick us up and how much it would cost us for the ride.

Whenever we think of systems where machine learning algorithms are used to make decisions, the most commonly considered framework that we might imagine is that a *machine learning model* is trained on some *dataset* that the learner has access to, and then the trained model is deployed in the world where it keeps making decisions. This setting does not consider a fundamental question: How does the learner get access to this *dataset*?

A more realistic situation in fact is that machine learning systems constantly interact with the environment they are deployed in and sequentially learn from the feedback they get about the decisions they make. A video-watching app would make recommendations to people using the app, but would also monitor which of the recommended videos people are actually watching to *learn* from the feedback and update itself. This is called the sequential decision-making problem where the learner has to make online decisions in real time and simultaneously learn from the sequentially arriving data to make better decisions in the future.

Classical literature on sequential decision-making has focused on understanding the problem by studying it across two axes. The first axis is the distribution of the sequential data, which could be either stochastic [2], that is coming from a fixed distribution that doesn't change with time, or the data could be adversarial, i.e. worst case in nature [3]. If an online ad exchange is trying to decide which advertisements to show to people based on how

likely an advertisement will get clicked, it is reasonable to assume that the person on the other side of the advertisement is a random sample from a population, thus is a stochastic sample from a fixed distribution. On the other hand, if a stockbroker is trying to decide whether to go long or short on a stock, the sequence of future stock prices is usually not stochastic across time and thus can be adversarial in nature.

The second axis along which the sequential decision-making problem is studied is based on the feedback the learner gets about their decisions. When an advertisement exchange decides to show one of the advertisements to a user, it only gets feedback about what the user thought about that particular advertisement that the exchange chose out of all possible advertisements. The learner cannot infer any information about what would have happened if they had chosen some other advertisement, thus the learner can only update their beliefs about the ad it chose Whereas for the stockbroker who is trying to decide whether to go long or short, even if they go long, they get to see the exact price of the stock on the next day, and calculate home much money they would have made if they had gone short instead. An even harder case is when the learner gets no feedback about their decisions or the feedback is very expensive to obtain. Consider an algorithm that is responsible for hate speech monitoring on a social media platform. After the algorithm blocks a post, in most cases it gets no feedback on whether it was right or wrong, and to obtain any feedback, the algorithm would have to take the help of a human reviewer who can actually classify if the content was violating some platform policy or not.

Increasingly, machine learning algorithms are now being deployed in complex socio-economic environments where they interact with agents and make sequential decisions that the agents may care about. Facebook and Google run millions of online ad auctions a day and multiple advertisers bid for multiple ad slots to show their ads to the users. Here the goal of the ad exchange, which is to maximize revenue, does not necessarily align with the goals of the advertisers. In such scenarios, if the learner or the ad exchange aims to learn from the actions of other agents so that it can make better decisions in the future, it has to take into

account the fact that the agents care about their own utilities. Thus agents participating in the system may behave *strategically* in hopes of changing the future decisions of the learner that are more favorable to them.

This has added a new dimension to the decision-making problem where the learner interacting with strategic agents must account for the fact their desirable outcomes may not necessarily be the desired outcomes for the agents present in the system. If the learner aims to learn from the agents in such a setting, then they have to design algorithms that are aware of the *incentives* of the strategic agents present in the system.

Let's formalize the setting where a learner makes sequential decisions while interacting with multiple agents.

1.1 Sequential Decision-Making with Agents

Let's consider a learner who is using an Algorithm named $\mathcal{A}$ to sequentially make decisions in an environment over T rounds while interacting with n agents.

In each round $t \in [T]$, the environment selects a true circumstance φ^t from a set of possible circumstances Φ. The circumstance φ^t denotes the true information about the current round t in which the learner has to make a decision, but the learner does not have access to φ^t. If the learner is trying to predict whether to go long or short on the stock at the beginning of day t, then $\varphi^t \in \mathbb{R}$ represents the change in the price of the stock by the end of the day t. If the learner is trying to decide which advertisement to select out of n possible ads with the aim of choosing advertisements that get clicked, then $\varphi^t \in \{0, 1\}^n$ represents the information if each possible ad $i \in [n]$ if selected in this round, would get clicked or not (that is $\varphi_i^t = 1$ or $\varphi_i^t = 0$).

Even though the learner doesn't have access to φ^t, the learner gets information by interacting with n agents who are present in the system. In each round t, for each agent, $i \in [n]$, the environment sets v_i^t representing agent i's belief about the circumstances in round t from the set of all possible beliefs $\mathcal{V}$. Recall the example of a learner trying to decide

what position to take on a stock, and let's assume they have access to n stock experts. Here each stock expert has a prediction $v_i^t \in \{1, -1\}$ representing the expert's recommendation to the stockbroker on going long or short on the stock. The stockbroker can use the advice of these experts to make decisions. On the other hand, in the online ad exchange problem if the ad exchange is auctioning the ad slots and these n agents are the advertisers who want to bid for these ad slots, then $v_i^t \in \mathbb{R}$ represents how much an advertiser i values the ad slot in round t. Thus $\boldsymbol{v}^t = (v_1^t, v_2^t, \cdots, v_n^t) \in \mathcal{V}^n$ represents the vector of beliefs of all agents in round t.

Instead of directly reporting the belief v_i^t to the learner, each agent i relays message b_i^t to the learner where b_i^t may or may not be equal to v_i^t. Here, $\boldsymbol{b}^t = (b_1^t, v_2^t, \cdots, b_n^t) \in \mathcal{V}^n$ represents the vector of reported beliefs of all agents in round t. In the ad exchange example, b_n^t represents the bid reported by an advertiser for the ad slot in round t, which could be different from how much they actually value the slot.

The algorithm $\mathcal{A}$ used by the learner can be defined as a function of all the information the learner has access to before the beginning of this round. Since the learner only has access to information from the previous rounds, the information available to the learner using algorithm $\mathcal{A}$ is represented by history $\mathcal{H}_{\mathcal{A}}^t \in \mathfrak{H}$ where $\mathfrak{H}$ is the set of all possible histories.

In each round t, the learner uses $\mathcal{A} : \mathfrak{H} \to \mathcal{P}$ to select a policy $\theta^t = \mathcal{A}(\mathcal{H}_{\mathcal{A}}^t)$ from the set of all possible policies $\mathcal{P}$. A policy $\theta \in \mathcal{P}$ is a function from the reported message space $\mathcal{V}^n$ of the n agents to the set of possible decisions $\mathcal{D}$ the learner can make in any round. The learner uses the selected policy θ^t to make the decision $\theta^t(\boldsymbol{b}^t)$ in round t.

In the stock market example, based on the historical performance of the different stock experts, the learner's policy θ^t represents how much money to invest in each stock expert's recommended plan; in the ad exchange example, the θ^t represents the function used by the learner to decide which advertiser wins the ad impression and how much they pay based on the reported bids $\boldsymbol{b}^t$.

At the end of the round t, the learner is rewarded $r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t) \in \mathbb{R}$ where $r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$ is the reward obtained by the learner in round t when they used policy θ^t to make the decision when the agents' belief was $\boldsymbol{v}^t$, the agents' reported message was $\boldsymbol{b}^t$, and the true circumstance was φ^t. The learner wants to maximize their total reward over all rounds t. In the stock market example, the reward is the total profit generated by the learner by investing money according to policy θ^t and in the ad exchange example, the reward is the total revenue generated by the ad exchange.

Other than the reward, the learner can also receive additional feedback about the policies it could have chosen in this round. The set of the complete feedback possible at round t is $\mathfrak{F}^t = \{r^t(\theta, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t) | \theta \in \mathcal{P}\}$, that is for each possible policy θ that the learner could have chosen, what would have been the reward obtained by the learner in the current round. Instead of observing the complete set $\mathfrak{F}^t$, the learner observes feedback $\mathcal{F}^t_{\mathcal{A}}$ which is a subset of $\mathfrak{F}^t$. In the *full-information* setting, the learner observes the full feedback in each round, that is $\mathcal{F}^t_{\mathcal{A}} = \mathfrak{F}^t$ for all $t \in [T]$. In contrast, if the learner observes only $\mathcal{F}^t_{\mathcal{A}} = \{r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)\}$, that is, it only receives feedback about the policy θ^t it chose and not about any other policy that it could have chosen, then it's called the *bandit-feedback* setting. For example, in the stockbroker example, since the learner can observe the real stock prices at the end of the day, they can also calculate how much reward it would have made if they had invested by following a policy $\theta \neq \theta^t$, whereas in the ad exchange example if the final payment received by the ad exchange depends on the fact if the ad got clicked or not, then the learner gets no information about the ads it did not choose. The feedback set $\mathfrak{F}^t$ in fact can in fact be empty, where the learner gets no feedback in most rounds, or it is expensive for the learner to obtain feedback. For example, in the case of online content moderation that we alluded to earlier, the learner requires a human reviewer to provide the correct label for the content under consideration which can be expensive. In such settings, the learner wants to minimize the label cost as well while maximizing the total reward.

Using the definition of $\mathcal{F}^t_{\mathcal{A}}$, we can formalize the history $\mathcal{H}^t_{\mathcal{A}}$ available to the learner in

round t. In the beginning at $t = 1$, the learner has no information, that is $\mathcal{H}^1_{\mathcal{A}} = \{\}$. In every round t, the learner observes the policy it chose θ^t, the reported bids $\boldsymbol{b}^t$, and the feedback it received $\mathcal{F}^t_{\mathcal{A}}$. Thus, for $t > 1$, the information available to learner before the start of round can be given by the recursive definition: $\mathcal{H}^t_{\mathcal{A}} = \mathcal{H}^{t-1}_{\mathcal{A}} \cup \{(\theta^{t-1}, \boldsymbol{b}^{t-1}, \mathcal{F}^{t-1}_{\mathcal{A}})\}$

Other than the learner, every agent i also obtains utility $u^t_i(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$ where $u^t_i(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$ is the utility of agent i in round t, when the when the agents' belief was $\boldsymbol{v}^t$, the agents' reported message was $\boldsymbol{b}^t$, and the true circumstance was φ^t. For the rest of the thesis, we use the words *learner*, *algorithm*, or $\mathcal{A}$ interchangeably to refer to the learner who is interacting with one of the n agents. We use the word *agent* to refer to one of the n agents interacting with the learner in the T round process. To summarize the setting, in each round t :

1. Environment sets true circumstance $\varphi^t \in \Phi$
2. Environment sets $\boldsymbol{v}^t = (v^t_1, v^t_2, \cdots, v^t_n) \in \mathcal{V}^n$ where $v^t_i \in \mathcal{V}$ is agent i's belief about the circumstances in round t
3. Simultaneously, the learner uses algorithm $\mathcal{A}$ to select policy $\theta^t = \mathcal{A}(\mathcal{H}^t_{\mathcal{A}}) \in \mathcal{P}$
4. Agents reports message $\boldsymbol{b}^t = (b^t_1, v^t_2, \cdots, b^t_n) \in \mathcal{V}^n$ to the learner where b^t_i is the message reported by agent i
5. Learner uses the policy to make decision $\theta^t(\boldsymbol{b}^t) \in \mathcal{D}$ and is rewarded $r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$
6. Every agent $i \in [n]$ obtains utility $u^t_i(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$
7. Learner observers feedback $\mathcal{F}^t_{\mathcal{A}} \subseteq \{r^t(\theta, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t) | \theta \in \mathcal{P}\}$

For measuring the performance of a learner using Algorithm $\mathcal{A}$, a standard notion of regret is used in sequential decision-making that compares the reward obtained by following the algorithm with the best possible reward the learner could have obtained by following a fixed policy θ in each round. That is, for a learner using algorithm $\mathcal{A}$ to make decisions for T rounds, the regret is given as follows.

$$\textsc{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t) - \min_{\theta \in \mathcal{P}} \sum_{t=1}^{T} r^t(\theta, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$$

A key desirable property we want is for the regret of the learner using algorithm $\mathcal{A}$, $\textsc{Reg}_{\mathcal{A}}(T)$ to be $o(T)$. This equates to the following:

$$\lim_{T \to \infty} \frac{\textsc{Reg}_{\mathcal{A}}(T)}{T} = 0 \tag{1.1}$$

If an algorithm satisfies Eq. (1.1), then it means that as $T \to \infty$, the average reward of the algorithm converges to the average reward of the optimal policy.

All the problems that we study in this thesis can be modeled as an instance of the general framework we described above. Here the learner wants to maximize their reward $r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$ over all rounds whereas agents care about their utilities $u_i^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)$ summed up over all rounds they participate in. If the decisions made by the learner in future rounds depend on the messages reported by the n agents in the current round, then the agents might be incentivized to behave strategically if leads to favorable outcomes in the future for them even at the cost of some utility for the current rounds.

The goal of this thesis is to analyze the sequential decision-making problem with multiple agents in socio-economic settings. Formally, we try to answer the following questions:

1. How can learners perform effective decision-making in the presence of strategic agents?

2. How can agents behave strategically to optimize for their own utility in sequential decision-making?

3. How much feedback is actually needed for the learner to effectively learn when the labels are expensive?

4. Can we make sequential decision-making algorithms that operate under limited feedback more *fair* in the decisions they make?

Thus, the contributions presented in this thesis can be divided into three themes:

- **Part I:** Sequential Decision-Making with Strategic Agents
- **Part II:** Sequential Decision-Making with expensive Feedback
- **Part III:** Sequential decision-making in a limited feedback environment while ensuring fairness

In the following sections, we summarize our contributions to the three settings and set up a roadmap for the thesis as well.

1.2 Sequential Decision-Making with Strategic Agents

In the sequential decision-making framework that we discussed in the last section, if the agents care about their own utilities, then they may aim to behave strategically to obtain more favorable outcomes. The choice of the policy θ^t chosen by the learner using algorithm $\mathcal{A}$ depends on the history $\mathcal{H}^t_{\mathcal{A}}$ available to the learner.

If the algorithm $\mathcal{A}$ uses the reported messages $\boldsymbol{b}^t$ of the agents and the feedback $\mathcal{F}^t_{\mathcal{A}}$ to select the policy θ^τ for rounds $\tau > t$, then the agents can potentially behave strategically in round t so that the algorithm selects more favorable policies for them in rounds $\tau > t$ even at the cost of losing some utility in the current round if it leads to a higher total utility. This creates extra challenges for the learner who now has to account for the strategic data it's receiving. In Part I of the thesis, we study sequential decision-making in the presence of strategic agents from the point of view of both the agents aiming to manipulate the learning algorithms and the system designers aiming to learn in presence of strategic agents.

There are two prominent forms of strategic gaming that the agents can employ in sequential decision-making settings to increase their individual payoff, potentially compromising the rewards of the learner.

First, agents can misreport their true beliefs, that is report message $b^t_i \neq v^t_i$, which is their true belief about the current round. If misreporting this information can lead to higher

individual payoffs then agents will possibly do so. If the algorithm $\mathcal{A}$ uses the reported messages $\boldsymbol{b}^t$, to select future policies, then agents who are aware of algorithm $\mathcal{A}$ being employed by the learner can report messages $\boldsymbol{b}^t \neq \boldsymbol{v}^t$ to influence the histories $\mathcal{H}_{\mathcal{A}}^{\tau}$ for $\tau > t$, aiming to influence the algorithm into making more favorable decisions for them.

Another attack aims to manipulate the feedback $\mathcal{F}_{\mathcal{A}}^t$ that the learner observes by corrupting the feedback using unfair means. For example, if an ad exchange is trying to select ads based on their average *clickiness*, an agent may create a robot that either clicks their own ads or does not click competing ads to adversarially *bias* the learning algorithm and make their own ad seem more desirable to the algorithm. Thus an adversarial agent can aim to influence the learner by changing the click results from $\varphi^t \in \{0,1\}^n$ to a corrupted $\tilde{\varphi}^t \in \{0,1\}^n$, and thus altering the learner's feedback from $\mathcal{F}_{\mathcal{A}}^t = \{r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \varphi^t)\}$ to $\tilde{\mathcal{F}}_{\mathcal{A}}^t = \{r^t(\theta^t, \boldsymbol{b}^t, \boldsymbol{v}^t, \tilde{\varphi}^t)\}$. This in turn changes the history of the learner $\mathcal{H}_{\mathcal{A}}^{\tau}$ for rounds $\tau > t$, hopefully leading to better outcomes for the adversarial agent.

We work with online ad auctions in an ad exchange as a running example for this theme of the thesis. We start with this work in Chapter 2, where we consider a revenue-maximizing auction in the repeated setting with strategic buyers. Consider an ad exchange sequentially auctioning ad impressions and multiple advertisers may participate in many of the rounds of the T round auction. In a single-round auction, the revenue-maximizing auction defined by Myerson [4] requires the auctioneer to have full knowledge of the true priors of the agents' values for the item up for auction. Since we are in the sequential setting, a learner may try to learn the prior aiming to run more revenue-generating auctions in the future. We show that by using tools from differential privacy for strategic robustness, we can limit the extra utility a learner can obtain from strategic manipulation. Thus, we design an algorithm where every agent can only gain a very small amount by trying to optimize for the optimal bid. Under this guarantee, if the bidders are willing to let go of the small extra utility, we are able to show a diminishing average revenue regret of the algorithm when compared to Myerson's auction, which is the per-round Bayesian revenue-maximizing auction. This chapter is based

on work published in Abernethy *et al.* [5].

In Chapter 3 we consider the second form of manipulation, where the agents use data poisoning attacks to manipulate a sequential learner. Consider an online ad auction where the ad exchange is deciding which ad to select based on how likely an ad gets clicked by a random user. This is an instance of a multi-arm bandit problem. We show that a large class of multi-arm bandit algorithms are vulnerable to adversaries who without even observing the arm chosen by the algorithm, an adversarial agent can make the algorithm suffer linear ($\Omega(T)$) regret by corrupting only a sublinear number of rounds. We also show that for many commonly used multi-arm bandit algorithms, the adversary can go a step further and by only corrupting a sublinear number of arms, it can make the algorithm choose an arm of the adversary's choosing for all but $o(T)$ rounds. This chapter is based on work published in Xu *et al.* [6].

In Chapters 2 and 3 we separately consider strategic agents who report their bids untruthfully to influence the learner or agents who instead of strategic bidding use unfair data corruptions attacks to influence a sequential decision algorithm into choosing decisions they prefer. In Chapter 4, we study pay-per-click ad auctions where agents bids for ad impressions but only pay if their ad impressions actually get clicked. Thus, agents can employ both a) *strategic bidding* to influence the learning outcomes and b) *strategic data poisoning attacks* to corrupt the feedback obtained by the learner by manipulating the click outcomes. We show that an *exploration separated* $\epsilon-$greedy style algorithm 1) is truthful, 2) recovers the $\tilde{O}(T^{2/3})$ lower bound in the absence of data corruptions, and 3) is robust to adversarial corruption attacks. This chapter is based on work presented in Abernethy *et al.* [7].

In Chapter 5, we take on the role of advertisers participating in online ad auctions and design bidding strategies in setting where the advertisers want to maximize their utility but are also constrained by a total budget on how much they can spend during the entire campaign. We give an end-to-end budget management system to show that in realistic

settings where the market conditions change with time, historical data can be used to plan out how the advertiser should spread their budget across time. We show that these plans can then be combined with bidding strategies from the stochastic setting to obtain vanishing regret in realistic non-stationary environments. This chapter is based on results presented in Kumar *et al.* [8].

1.3 Sequential Decision-Making with expensive Feedback

In Part II, we switch gears, and instead of strategic feedback, we consider the settings where the feedback about the learner's decisions is hard to obtain. We consider the classical problem of multiclass prediction with expert advice, but with an active learning twist in Chapter 6. The classic algorithm for online learning with expert advice is commonly known as *Hedge* [9], although variants are often referred to as *exponential weights* or *weighted majority* [10].

One of the downsides of Hedge, as with many online learning algorithms, is that it is not *label efficient*: the learning process requires that we observe the target y^t on each round. Obtaining individual labels can, quite often, be very expensive to the learner; indeed this is central to why we design prediction algorithms in the first place.

Active learning, which refers broadly to a family of frameworks in which the learning algorithm can make selective label queries, are designed precisely with the goal of minimizing the number of needed labels while achieving a suitable learning performance[11, 12, 13, 14, 15, 16, 17, 18]. The key idea is that we do not necessarily need to have a batch of labeled examples prior to training, in many natural scenarios the algorithm may be able to actively engage with the labeling process to query labels on a set of unlabelled examples. It is worth noting up front that nearly all work on active learning has imagined a *batch* setting, where the algorithm is evaluated only at the end of the learning process, in expectation, on new samples. This is surprising, in particular, given that active learning methods are by their nature online, as they seek to iteratively refine their learning process and selection of

samples. But thus far there has been no work on putting active learning algorithms to the test in a no-regret setting of prediction with expert advice, where the algorithm's decision is evaluated at each round of the sequence, and where the expert's predictions, as well as the labels, can be non-stochastic and potentially chosen by an adversary.

We aim to remedy this gap and show that there is a natural framework for active learning in the no-regret setting of prediction with expert advice with strong learning guarantees as well as bounded label complexity. First, we define a notion of complexity of the experts' predictions, somewhat akin to the disagreement coefficient, that provides a key tool in obtaining a provable guarantee; we refer to this as *compactness* for a parameter $\zeta \geq 1$ that measures the *active learnability of the prediction matrix*. The ζ compactness of the prediction matrix is closely related to the *disagreement coefficient* considered in batch active learning framework [19, 11] and can be thought of as a combinatorial counterpart in the online setting. We give an algorithm that is an *active* version of Hedge and obtains the exact same regret as Hedge, but only uses $O(\zeta L^*)$ labels where L^* is the number of mistakes made by the best expert, compared to the (T) labels queried by Hedge. The results presented in Chapter 6 have been published in Kumar *et al.* [20].

1.4 Sequential decision-making in a limited feedback environment while ensuring fairness

In Part III of the thesis, we consider a multi-arm bandit setting where instead of just optimizing for the total reward over all rounds, the learner wants to make decisions that also satisfy some notion of *fairness*. Wang *et al.* [21] introduce a multi-arm bandit setting where the goal of the learner is to select each agent with probability proportional to a monotonically increasing function of their mean reward. For example, consider an algorithm that is deciding which products to rank higher in search results on an online shopping platform. If two products have a similar quality but one is slightly better than the other, then a traditional multi-arm bandit algorithm aimed at maximizing the reward will always select

the product with slightly higher quality, whereas it would be fairer to give some exposure to the slightly worse product too with a probability proportional to how good the product actually is. We show that if arms belong to a certain group, i.e. being products of the same company, then agents can game the fairness notion given in Wang *et al.* [21] and add near duplicates products to their lineup to increase their total exposure. To counter this, we introduce a group-based notion of fairness that says each group should also be selected proportional to a function of the quality of the arms in the group. This notion limits the kind of manipulation possible by duplication. We propose a new notion of *fairness* regret that promotes fairness of exposure to both groups and individual arms show that an ϵ-greedy style algorithm obtains $O(T^{2/3})$ fairness regret. This chapter is based on an ongoing project [22].

Part I

Sequential Decision-Making with Strategic Agents

CHAPTER 2
REVENUE MAXIMIZATION IN REPEATED AUCTIONS WITH STRATEGIC BIDDERS

In this chapter, we study the problem of learning in repeated auctions to maximize revenue. The classical approach to maximizing revenue requires a known prior distribution on the demand of the bidders, although recent work has shown how to replace the knowledge of a prior distribution with a polynomial sample. However, in an online setting, when buyers can participate in multiple rounds, standard learning techniques are susceptible to *strategic manipulation*: bidders can improve their long-term wellbeing by manipulating the trajectory of the learning algorithm through bidding. For example, they may be able to strategically adjust their behavior in earlier rounds to achieve lower, more favorable future prices. Such non-truthful behavior can hinder learning and harm revenue. In this chapter, we show how tools from differential privacy, mechanism design, and sample complexity can be combined to give a repeated auction that (1) learns from bidders' past bids, (2) is approximately revenue-optimal, and (3) strategically robust, as it incentivizes bidders to behave truthfully. This chapter is based on work published in Abernethy *et al.* [5].

2.1 Introduction

When we observe prices in market settings—stock exchanges, farmers' markets, ad auctions—we understand that these prices were not chosen arbitrarily. Rather, the seller (auctioneer, market maker, etc.) selected these prices after observing a stream of previous transactions, which provide relevant information about the demands of buyers that are key to maximizing income as well as managing available inventory. The process of setting prices from a growing database of previous sales is fundamentally a learning problem, with all of the typical trade offs akin to bias versus variance, etc. However, in the case of repeated auctions,

there is one additional challenge: market participants are often quite aware of the underlying learning procedures employed by the auctioneer and can seek to benefit using deceptive bidding strategies. Buyers, in other words, can act strategically to influence the learning procedures which could lead to more favorable outcomes for them. The agents can therefore lead the algorithm into *strategic overfitting* introducing additional hurdles to learning problem at hand.

Consider the example where an online ad exchange is selling T ad impressions sequentially by running T auctions. In each round n advertisers who are interested in showing theirs ads to the users participate in auction and bid for the impression based on how much they value it. Advertisers are also interested in multiple ad impressions so they may participate in multiple rounds of the auction. In this case, the ad exchange or the auctioneer wants to maximize the total revenue they obtain over the T rounds, but advertisers only care about their own their own total utility over all the rounds of auction they participate in. The goals of the learner and the agents don't necessarily match. If the auctioneer plans on learning from the reported bids of the agents, hoping to run better revenue generating auctions in the future, the agents can take advantage of the situation and bid *untruthfully* aiming to increase their own total utility instead.

Under Bayesian assumptions where agents only act once, auction pricing has been well understood since the work of Myerson [4], who characterized the truthful revenue-optimal scheme and showed that it is a function of the prior distribution of how much bidders *value* the item in auction. Thus, if an auctioneer aims to obtain expected revenue close to Myerson [4]'s auction, it needs to learn information about the prior distributions of agents' values.

Frequentist alternatives to this model have been introduced in recent years [23, 25, 26, 27, 28, 29, 30, 31, 32, 24], with the goal of designing auctions with good revenue guarantees if one does not have a prior but instead is given only samples from the underlying distribution. These methods, however, still imagine only a one-shot mechanism, and are not robust to multi-round strategic behavior of bidders.

In this chapter, we design multiround auction-learning algorithms that exhibits theoretical guarantees that limit a buyer's ability to manipulate the mechanism towards their own benefit. Our results aim to nudge the development of optimal auctions closer to realistic environments where such mechanisms are deployed. We employ tools from *differential privacy* as our core technique to control the impact of any individual buyer's strategy on her utility in future participation. A differentially private mechanism ensures that that the output of a computation has only a small dependence on any one input data point.

Privacy has previously been used as a tool to achieve truthfulness in a variety of game theoretic environments [33], including mechanism design [34, 35], mediated games [36, 37], and market design [38, 39, 40, 41]. Our seller's learning algorithm is differentially private with respect to bid data, which limits the effect of each player's bid on future choices of single-round auctions, thus disentangling incentives across rounds. In this sense, we use differential privacy not as a tool for information security but instead for robustness; this, in turn, yields the desired incentive guarantees.

Summary of our results Our main contribution in this chapter is the first computationally tractable algorithm for learning nearly-optimal Bayesian revenue-maximizing auctions with an approximate truthfulness guarantee in a repeated setting. We are able to show the following:

Theorem 2.1.1 (Informal)**.** *We give a learning algorithm $\mathcal{A}$ which guarantees that*

1. *there exists an approximate equilibrium in which all bidders report their true values. That is, assuming all bidders behave truthfully, a agent can obtain only $O(\epsilon)$ total extra utility in future rounds by deviating from the truthful bidding a single round, and*

2. *under truthful bidding, with probability at least $1-\alpha$, the average expected revenue of $\mathcal{A}$ is*

$$\textit{Rev} \geq \textit{OPT} - \beta - n^2\tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{T\epsilon}\right)$$

where OPT is optimal expected revenue of Myerson [4] in a single round, β is a small discretization parameter that can be set as $o(T)$

Along the way to this result, we provide several useful technical lemmas for comparing the revenue of mechanisms on two similar distributions, and comparing the revenue of two similar mechanisms on any fixed distribution which may be of independent interest.

Before we expand on these results and provide the technical details or our contributions, we formally introduce the setting under consideration in 2.2 and provide a quick background on mechanism design in Section 2.2.3 which will help us formalize what it means for a bidder to behave strategically, or why a bidder may behave strategically, and what it means to design algorithms which can deter strategic behavior.

2.2 Model and Preliminaries

We consider a T round auction, where in each round, the seller or auctioneer $\mathcal{A}$ is auctioning a single item to n bidders. Each bidder participating in this auction is interested in one copy of the of the item in each round but can participate in future rounds as well. In each round t, the following interactions occur between the seller $\mathcal{A}$ and the bidders:

1. The environment sets $\boldsymbol{v}^t = (v_1^t, v_2^t, \cdots, v_n^t)$ where for each $i \in [n]$, $v_i^t \in \mathcal{V} \subseteq \mathbb{R}$ is sampled from a fixed distribution D_i and represents the amount that the bidder i is willing to pay for the item, or the value for the bidder i in round t

2. The seller $\mathcal{A}$ selects an auction mechanism $\mathcal{M}^t := (x^t, p^t)$ where $x^t : \mathcal{V}^n \to \mathcal{X}$ is an *allocation rule* which takes in a vector of bids $\boldsymbol{b}$ and returns a feasible allocation of the items, where $x_i^t(\boldsymbol{b})$ is 1 if the bidder i receives the item and 0 otherwise. Similarly $p^t : \mathcal{V}^n \to \mathbb{R}^n$ is a *payment rule*, which takes the bid profile $\boldsymbol{b}$ and outputs a vector of payments demanded of each player

3. Bidders report bids $\boldsymbol{b}^t = (b_1^t, b_2^t, \cdots, b_n^t)$ where $b_i^t \in \mathcal{V} \subseteq \mathbb{R}$ is the bid reported by bidder i

4. The seller decides who all won the auction using the allocation $x^t(\boldsymbol{b}^t)$ where $x_i^t(\boldsymbol{b}^t)$ is 1 if i receives the item and 0 otherwise. Each bidder i is also charged a payment $p_i^t(\boldsymbol{b}^t)$ regardless of whether they won the item or not.

As a conclusion of each round t, the revenue obtained by the seller in round t is

$$r(\mathcal{M}^t := (x^t, p^t), \boldsymbol{b}^t, \boldsymbol{v}^t) = \sum_i^n p_i^t(\boldsymbol{b}^t)$$

The utility obtained by an agent i is the value v_i^t if they wins the item minus the price $p_i^t(\boldsymbol{b}^t)$ they has to pay regardless, that is,

$$u_i(\mathcal{M}^t := (x^t, p^t), \boldsymbol{b}^t, \boldsymbol{v}^t) = v_i^t \cdot x_i^t(\boldsymbol{b}^t) - p_i^t(\boldsymbol{b}^t)$$

Since the seller is limited to selling J items in each round, the feasible set of allocations $\mathcal{X}$ is $\{x \in \{0,1\}^n,\ \|x\|_1 <= J\}$. We let $\mathbf{D} = D_1 \times \cdots \times D_n$ denote the product distribution of value distributions, and we use $\boldsymbol{v}$ to denote a vector of values sampled from this distribution. Furthermore, we let $\boldsymbol{v}_{-i}$ denote $\boldsymbol{v}$ with the i-th element removed, and use $(v_i', \boldsymbol{v}_{-i})$ to denote the same vector with v_i' replacing the i-th element. We also assume that the values and bits in each round are bounded between 0 and h, i.e $\mathcal{V} = [0, h]$

We say that an agent i behaves *truthfully* if they report their true value in each round to the learner, i.e. $b_i^t = v_i^t$. In this setting, the seller aims to maximize their total revenue in all T rounds of auctions, while each agent wants to maximize their total utility from all the rounds in which they participate. If the choice of mechanism $\mathcal{M}^\tau$ made by the algorithm in round $\tau > t$ is a function of bidder's reports $\boldsymbol{b}^t$, it may incentivize bidders to behave *strategically* and misreport their bids, $b_i^t \neq v_i^t$, if leads to better outcomes for them in the future. Thus, any algorithm must account for the *incentives* of the bidders that may lead them to report strategically and make the algorithm's learning process harder.

In fact, even in the absence of any learning, strategic agents may have incentives to misreport their bids to maximize their own utility. In the following subsection, we introduce

preliminaries about mechanism design starting from a single round auction and extending the ideas to repeated auctions that help us formalize some of these notions.

The setting under consideration is similar to Liu *et al.* [42] where a bidder from any population may appear several times over the course of the T rounds, drawing a fresh value each time. In this setting, bidders may have an incentive to misreport their values in order to change the mechanism in future rounds, and their potential reward for doing so depends on the number of future rounds in which they expect to participate. Amin *et al.* [43] show that very little can be done when a bidder participates in every round, so we assume this cannot occur. Formally:

Assumption 2.2.1. *No bidder participates in more than k rounds of the T-round auction.*

2.2.1 Mechanism Design Basics

Let's start with a single round auction with n bidders. As discussed earlier, one can view a *mechanism* (auction) $\mathcal{M} := (x, p)$ as having two components:

- A possibly randomized *allocation rule* $x : \mathcal{V}^n \to \mathcal{X}$, which takes in a vector of (bids) $\boldsymbol{b}$ and returns a feasible allocation of the items, where $x_i(\boldsymbol{b})$ is 1 if i wins the item and 0 otherwise; and
- A *payment rule* $p : \mathcal{V}^n \to \mathbb{R}^n$, which takes $\boldsymbol{b}$ and returns a vector $p(\boldsymbol{b})$ where $p_i(\boldsymbol{b})$ is the payment to charged to bidder i.

The protocol followed in a single-round auction is the following:

1. The seller selects a mechanism $\mathcal{M} := (x, p)$ that is public to the bidders
2. Bidders select strategy $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_n)$ as a response to $\mathcal{M}$ where $\sigma_i : \mathcal{V} \to \mathcal{V}$ is the strategy bidder i will use to calculate their bid as a function of their value.
3. The environment sets bidders' values $\boldsymbol{v} = (v_1, v_2, \cdots, v_n) \sim \mathbf{D}$

4. Bidders report bid $\boldsymbol{b} = \boldsymbol{\sigma}(\boldsymbol{v}) := (\sigma_1(v_i), \cdots, \sigma_n(v_n))$

The auctioneer's revenue from the mechanism is

$$r(\mathcal{M} := (x, p), \boldsymbol{b}, \boldsymbol{v}) = \sum_{i}^{n} p_i(\boldsymbol{b}) \tag{2.1}$$

We make the standard assumption that the bidders have *quasi-linear utility*: for a vector of bids $\boldsymbol{b}$ (which may not necessarily match the values $\boldsymbol{v}$), bidder i's utility for allocation $x(\boldsymbol{b})$ and payment $p(\boldsymbol{b})$ is

$$u_i(\mathcal{M} := (x, p), \boldsymbol{b}, \boldsymbol{v}) = v_i \cdot x_i(\boldsymbol{b}) - p_i(\boldsymbol{b}) \tag{2.2}$$

As the auctioneer has to make the Mechanism $\mathcal{M}$ public as part of rules of the game, the bidders can choose their strategy $\boldsymbol{\sigma}$ as a response to $\mathcal{M}$.

Another measure of importance is the social welfare of the auction that is defined as the total sum of utilities of all bidders and the revenue of the seller. Formally, the welfare of an auction $\mathcal{M}$ is:

$$w(\mathcal{M} := (x^t, p^t), \boldsymbol{b}^t, \boldsymbol{v}^t) = \sum_{i}^{n} v_i^t \cdot x_i^t(\boldsymbol{b}^t) \tag{2.3}$$

In this chapter, we focus on revenue maximization but we will discuss more about welfare maximization in Chapter 4.

We now introduce the notion of a *truthful* mechanism. In mechanism design, a truthful mechanism is also called an incentive-compatible mechanism, and there are different notions of incentive compatibility based on how strict the requirements are. Let's define one of the strictest but but simple notions of truthfulness called Dominant Strategy Incentive Compatible.

Definition 2.2.1 (Truthful mechanism (DSIC))**.** A mechanism $\mathcal{M}$ is dominant strategy incentive compatible if for every agent i, for any strategy profile $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_n)$, for any

value profile $\boldsymbol{v}$, it must hold that

$$u_i(\mathcal{M}, (v_i^t, \boldsymbol{\sigma}_{-i}(\boldsymbol{v}_{-i}), \boldsymbol{v}) \geq u_i(\mathcal{M}, \boldsymbol{\sigma}(\boldsymbol{v}), \boldsymbol{v})$$

In simpler words, Definition 2.2.1 states that a mechanism $\mathcal{M}$ is truthful if for every bidder i, irrespective of the strategy followed by other bidders, it is always in the bidders best interest to bid $b_i = v_i$. If an auction $\mathcal{M}$ is not truthful, the bidders may follow a strategy where $b_i = \sigma_i(v_i) \neq v_i$.

Since we are in the Bayesian setting where the values $\boldsymbol{v}$ are sampled from a fixed distribution $\mathbf{D}$, assuming for the moment that the bidders bid their true values $\boldsymbol{v}$, we can define the *expected revenue* of a mechanism $\mathcal{M}$ as the expectation of the payments received,

Definition 2.2.2 (Expected Revenue)**.** The expected revenue of a Mechanism $\mathcal{M}$ with bids sampled from $\mathbf{D}$ is

$$\text{Rev}(\mathcal{M}; \mathbf{D}) := \mathbf{E}_{\boldsymbol{v} \sim \mathbf{D}}[\textstyle\sum_{i=1}^{n} p_i(\boldsymbol{v})]$$

Let us now recall a classical result in Bayesian-optimal mechanism design when the seller's goal is to maximize revenue. Myerson [4] essentially fully characterized the solution in this setting. The interested reader can learn more in Hartline [44]; we briefly review these results here in two pieces. The first piece states that payments in truthful mechanisms essentially depend solely on the allocation function.

Theorem 2.2.2 (Payment Identity, Myerson [4])**.** *A mechanism $\mathcal{M}$ is truthful (DSIC) if and only if for all bidders i and bid profile $\boldsymbol{b}$*

1. *It has a monotone allocation rule, that is $x(b_i, \boldsymbol{b}_{-i})$ is monotone non decreasing in b_i and*

2. *Payments satisfy $p_i(\boldsymbol{b}) = b_i \cdot x_i(b_i, \boldsymbol{b}_{-i}) - \int_0^{b_i} x_i(z, \boldsymbol{b}_{-i}) dz + p_i(0, \boldsymbol{b}_{-i})$*

The second key result is that for truthful mechanisms, the expected revenue can be written in terms of welfare in a remapped *virtual value* space. We overload the notation for

a distribution and $D_i(x)$ represents the cumulative density function of D_i at x, and $d_i(v_i)$ denotes the probability density function.

Theorem 2.2.3 (Myerson [4])**.** *For any truthful mechanism $\mathcal{M} = (x, p)$ with bidder values distributed according to* $\mathbf{D}$*, the expected revenue from player i can be written as $E_{\mathbf{v}\sim\mathbf{D}}[\phi_i(v_i)x_i(\mathbf{v})]$, where $\phi_i(v_i)$ is the* virtual value*, given by $\phi_i(v_i) = 1 - \frac{1-D_i(v_i)}{d_i(v_i)}$. So, Rev$(\mathcal{M}; \mathbf{D}) = \mathbb{E}_{\mathbf{v}\sim\mathbf{D}}[\sum_i \phi_i(v_i)x_i(\boldsymbol{v})]$.*

We will use the notation $\mathcal{M}^*_{\mathbf{D}}$ to denote the revenue-optimal mechanism for distribution $\mathbf{D}$,—Myerson provides a precise construction of this auction.

Definition 2.2.3 (Myerson's Auction)**.** Fixing a prior distribution $\mathbf{D}$, given a value profile $\boldsymbol{v}$ Myerson's revenue-optimal mechanism $\mathcal{M}^*_{\mathbf{D}}$ calculates virtual values $\phi_i(v_i) = \phi_i(v_i) = 1 - \frac{1-D_i(v_i)}{d_i(v_i)}$ and (a) selects the feasible allocation which maximizes virtual welfare $[\sum_i \phi_i(v_i)x_i(\boldsymbol{v})]$ according to the virtual values and (b) charges payments according to the Payment Identity of Theorem 2.2.2.

From Theorem 2.2.2, we know that for a truthful auction, allocation function $x(b_i, \boldsymbol{b}_{-i})$ should be monotonically non-decreasing in b_i. Thus, to implement Myerson's allocation rule truthfully, we require that the virtual value function for each agent i, that is, $\phi_i(v) = 1 - \frac{1-D_i(v)}{d_i(v)}$ should be a monotonically non-decreasing function of v. This is in general true for a large class of distributions, called regular distributions.

Assumption 2.2.4. *For each population $i \in [n]$, the value distribution D_i is regular, that is, the virtual value function $\phi_i(v) = 1 - \frac{1-D_i(v)}{d_i(v)}$ is a monotone non decreasing function of v.*

Regularity of value distributions is a standard assumption in auction design ([4]), but even when the distributions are not regular, Myerson's auction can be implemented by calculating *ironed virtual values* such that the allocation rule becomes monotone while maximizing the true virtual welfare. We refer the reader to Hartline [44, Chapter 3] for a detailed exposition on this topic.

This implies that the optimal expected revenue for a truthful auction is $\text{Rev}(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D})$ where $\mathcal{M}^*_{\mathbf{D}}$ is the Myerson's auction described by Definition 2.2.3. An important conclusion from this discussion is that the truthful expected revenue-maximizing auction, i.e. Myerson's auction required the seller to know information about the exact prior $\mathbf{D}$ of the bidder's values. Several alternatives to this model have been introduced in recent years [23, 25, 26, 27, 28, 29, 30, 31, 32, 24], with the goal of designing auctions with good revenue guarantees if one does not have a prior but instead only samples from the underlying distribution are given. These methods *learn* use these samples to select mechanism $\mathcal{M}$, however, they still imagine only a one-shot mechanism, assuming all the bids in their dataset are truthful reports. Thus, these methods are not robust to the multi-round strategic behavior of bidders.

Since the seller aims to maximize their total expected revenue, we can generalize the notion of expected revenue given in Definition 2.2.2 to account for *untruthful* mechanisms as well. Let us assume that a single round mechanism is $\mathcal{M}$ which may or may not be truthful, and true values are sampled from $\mathbf{D}$, and each agent uses a fixed strategy σ_i. This is equivalent to the bidder reporting bids which are samples from another distribution F_i where a sample b_i from F_i is generated by sampling v_i from D_i and returns $\sigma_i(v_i)$. Thus, for a mechanism $\mathcal{M}$ with bids $\boldsymbol{b}$ being sampled from the bid distribution $\boldsymbol{F} = F_i \times \cdots \times F_n$, the expected revenue can be written as $\text{Rev}(\mathcal{M}; \boldsymbol{F}) := \mathbf{E}_{\boldsymbol{v} \sim \boldsymbol{F}}[\sum_{i=1}^{n} p_i(\boldsymbol{v})]$

The mechanism design preliminaries discussed so far are for *one shot* games where players do not observe the past actions of others and adjust their strategy accordingly. We now turn our attention to multi-round play; we need to expand our notion of player behavior and strategy. In the multi-round auction setting, we described at the beginning of this section, in each round t the Algorithm $\mathcal{A}$ selects a new auction $\mathcal{M}^t$ possibly as a function of their observed history $\mathcal{H}^t_{\mathcal{A}}$ which includes all the reported past bids ($\{\boldsymbol{b}^\tau; \tau \leq t\}$) and the choice of mechanisms the algorithm made in the past ($\{\mathcal{M}^\tau; \tau \leq t\}$). In the multi-round auction protocol, the exact auction $\mathcal{M}^t$ that will be selected in each round is not public to the bidders at the beginning of the complete T-round auction, but how the algorithm $\mathcal{A}$ selects the

auctions is made public. In our setting, the learner starts with no prior information about the bidders value distributions, thus to maximize revenue $\mathcal{A}$ has to learn from the bidder's report so that it runs better auctions in the future rounds and obtain more expected revenue.

In response to the choice of algorithm $\mathcal{A}$, the bidders will now possibly follow an adaptive strategy $\boldsymbol{\sigma} = \{\sigma_i^t, i \in [n], t \in [T]\}$, such that the strategy followed by agent i in round t, σ_i^t not only is a function of the current value v_i^t, but also accounts for the history $\mathcal{H}_i^t$ of their observations. We assume that the agents observe their own outcomes $x_i(\boldsymbol{b}^t)$ and $p_i(\boldsymbol{b}^t)$ in the rounds in which they participate, but not the full historical data used by the designer to produce the mechanism each round. Thus, the history $\mathcal{H}_i^t$ in round t for each agent i then consists of all their reported bids b_i^τ, their own allocation $x_i^t(\boldsymbol{b}^\tau)$, and their payments $p_i^t(\boldsymbol{b}^\tau)$ for each round $\tau \leq t$ in which they participated previously. Given a history $\mathcal{H}_i^t$ and a value v_i^t in the current round t for agent i, we denote the bid of agent i using a strategy σ_i^t as $\sigma_i^t(v_i^t\,;\,\mathcal{H}_i^t)$. We suppress the dependence on history when clear from context and denote the bid of bidder i in round t using strategy σ_i^t as $\sigma_i^t(v_i^t)$.

The repeated auction version of Dominant Strategy Incentive Compatibility (Definition 2.2.1) would imply that for every agent i, for any realization of values for values $\boldsymbol{v}^1, \cdots, \boldsymbol{v}^T$ and competing bids $\boldsymbol{b}_{-i}^t$, the agents total utility is maximized by bidding truthfully in every round. Since, we are in a Bayesian setting where the values are being sampled in a stochastic manner, we consider a Bayesian version of equilibrium and incentive compatibility.

Informally, using an algorithm $\mathcal{A}$, a profile of strategies $\boldsymbol{\sigma}$ is an *equilibrium* for this game if in every round t, for every agent i, and every history $\mathcal{H}_i^t$ that agent i might have observed previously, if all agents follow strategy $\boldsymbol{\sigma}$, then agent i's strategy in round t, σ_i^t maximizes their expected total utility over the current and future rounds. Here the expectation is taken over the randomness of agent i's beliefs about other agents, as well as the future values of all agents. Formally, let $U_i^t(\boldsymbol{\sigma}, v_i^t, \mathcal{H}_i^t)$ denote the total expected utility of agent i in rounds $\tau > T$ when every bidder follows strategy $\boldsymbol{\sigma}$, given their value of v_i^t in the current round

and their observed history $\mathcal{H}_i^t$. Note that since we make the assumption that agent i in round T is from a population characterized by D_i, but each particular agent only participates in k of the T round auction (Assumption 2.2.1). Let $C_i^t \subseteq [T]$ be the set of auctions that the agent i from round t also participates in. Assumption 2.2.1 implies that for every $i \in [n]$ and $t \in [T]$, $|C_i^t| \leq K$. Thus the agent i in round t only sums over the utilities from the rounds it will participate in to calculate future utility $U_i^t(\boldsymbol{\sigma}, v_i^t, \mathcal{H}_i^t)$.

$$U_i^t(\boldsymbol{\sigma}, v_i^t, \mathcal{H}_i^t) = E_{[\boldsymbol{v}_{-i}^t, \boldsymbol{v}^\tau, \cdots, \boldsymbol{v}^T]}[\sum_{\tau \geq t \cap C_i^t} u_i(\mathcal{M}^\tau, \boldsymbol{\sigma}^\tau(\boldsymbol{v}^\tau), \boldsymbol{v}^\tau)] \tag{2.4}$$

Note that in Eq. (2.4), the expectation is still taken over all round $\tau \geq t$ and over the private values of other bidders in this round $\boldsymbol{v}_{-i}^t$ to account for the expected behavior of the algorithm and the other agents that participate in the auction.

Let $(\tilde{\sigma}_i^t, \boldsymbol{\sigma}_{-(i,t)})$ be the strategy when the agent i deviates from the strategy $\boldsymbol{\sigma}$ in round t, for all other agents and rounds, the strategy followed is $\boldsymbol{\sigma}$. The total expected utility of the agent i from the current and all future rounds now becomes

$$U_i^t((\tilde{\sigma}_i^t, \boldsymbol{\sigma}_{-(i,t)}), v_i^t, \mathcal{H}_i^t) = E_{[\boldsymbol{v}_{-i}^t, \boldsymbol{v}^{t+1}, \cdots, \boldsymbol{v}^N]}[u_i(\mathcal{M}^t, (\tilde{\sigma}_i^t(v_i^t), \boldsymbol{\sigma}_{-i}^\tau(\boldsymbol{v}_{-i}^t)), \boldsymbol{v}^t) + \textstyle\sum_{\tau > t \cap C_i^t} u_i(\mathcal{M}^\tau, \boldsymbol{\sigma}^\tau(\boldsymbol{v}^\tau), \boldsymbol{v}^\tau)]$$

Here we have suppressed the dependence on history $\mathcal{H}_j^\tau$ for each agent $j \in [n]$ and rounds τ and the history $\mathcal{H}_\mathcal{A}^\tau$ for the algorithm $\mathcal{A}$ where the algorithm $\mathcal{A}$ selects the mechanism $\mathcal{M}^t$ in round t as a function of it's observed history $\mathcal{H}_\mathcal{A}^\tau$. It is implied that the expected future utility calculated by agent i takes into account whatever the observed histories could be for other agents and algorithm $\mathcal{A}$. Changing the strategy of agent i even in a single round t, can affect the observed histories h_j^τ of all agents $j \in [n]$ (including i) and the observed history of the algorithm $\mathcal{H}_\mathcal{A}^\tau$ for future rounds $\tau \geq t$, thus affecting the decision made by other bidders and the algorithm in future rounds. Therefore, when calculating their expected utility, the learner must consider that changing their strategy in a single round can change the future bids of other bidders and the mechanisms selected by algorithm $\mathcal{A}$ even if the

other bidders have committed to a strategy in advance.

Definition 2.2.4 (Perfect Bayesian Nash Equilibrium (BNE))**.** A profile of strategies $\boldsymbol{\sigma} = \{\sigma_i^t; i \in [n], t \in [T]\}$ is an η*-approximate Perfect Bayesian equilibrium* if for every agent i, round t, history $\mathcal{H}_i^t$, value v_i^t, given that strategy $\boldsymbol{\sigma}_{-(i,t)}$ is followed by all other agents and in all rounds, then $\sigma_i^t(v_i^t\,;\,\mathcal{H}_i^t)$ approximately maximizes agent i's total expected utility from future rounds up to an additive η. That is $U_i^t(\boldsymbol{\sigma}, v_i^t, \mathcal{H}_i^t) \geq U_i^t((\tilde{\sigma}_i^t, \boldsymbol{\sigma}_{-(i,t)}), v_i^t, \mathcal{H}_i^t) - \eta$ for any alternate strategy $\tilde{\sigma}_i^t$. If $\eta = 0$, we say that $\boldsymbol{\sigma}$ is an *exact Perfect Bayesian Equilibrium*.

This implies that a strategy $\boldsymbol{\sigma}$ in Perfect BNE if for every bidder i and for every history of the game, if all other bidders besides i behave according to the strategy $\boldsymbol{\sigma}$, then playing σ_i^t is expected utility-maximizing behavior for the bidder i in round t. On the other hand, an η-approximate Perfect Bayesian equilibrium implies that if an agent i deviates from strategy σ_i^t, then they can get at most η extra utility in and future rounds. Using the definition of the approximate equilibrium, we now define a notion of approximate truthfulness or incentive compatibility for the multi-round auction.

Definition 2.2.5 (η-utility-approximate BIC)**.** An algorithm $\mathcal{A}$ is η *utility-approximately Bayesian incentive compatible* if the strategy profile where every agent bids truthfully in every history is an η-approximate Perfect Bayesian equilibrium.

Definition 2.2.5 of η-utility-approximate BIC guarantees that all bidders bidding truthfully in all rounds is an (approximate) Bayes-Nash equilibrium (BNE). In other words, if an agent deviates from truthful bidding in any round t and tries to bid strategically, then they only gain an extra η compared to the utility they would have obtained if they had bid truthfully in this round. It does not make any statement about how far the optimal bid could be from the truthful bid if the agents do, in fact, behave strategically.

Using the definition for expected revenue (Definition 2.2.2) for a single round mechanism, if using algorithm $\mathcal{A}$ that selects the mechanism $\mathcal{M}^t$ in round t, and if the strategic behaviour of bidders results in bid distributions $\boldsymbol{F}^t$ in round t, then total expected revenue of the

algorithm $\mathcal{A}$ can be given by

$$\text{Rev}_{\mathcal{A}}(T) = \sum_{t=1}^{T} \text{Rev}(\mathcal{M}^t; \boldsymbol{F}^t)$$

To measure the performance of our algorithm, we use a standard notion of expected regret that compares the average expected revenue of our algorithm $\mathcal{A}$ with the expected per round revenue of a revenue optimal auction that knows the distribution $\mathbf{D}$. Here the benchmark corresponds to Myerson's auction which is the revenue maximizing truth auction in the bayesian setting.

Definition 2.2.6 (Expected Regret)**.** For a given algorithm $\mathcal{A}$, that selects mechanism $\mathcal{M}^t$ in round t, given that the true bidder values are sampled from $\mathbf{D}$ and the bidders behave strategically, resulting in a bid distribution $\boldsymbol{F}^t$ in round t, the expected regret of $\mathcal{A}$ over all T rounds is given by

$$\textsc{Reg}_{\mathcal{A}}(T) = T \cdot \text{Rev}(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D}) - \sum_{t=1}^{T} \text{Rev}(\mathcal{M}^t; \boldsymbol{F}^t)$$

where $\mathcal{M}^*_{\mathbf{D}}$ is the Myerson's auction described by Definition 2.2.3.

2.2.2 Related works

Liu *et al.* [42] study the same problem of revenue maximization in repeated auctions where bidders may appear more than once but only optimize over simpler class of mechanisms like posted prices or anonymous reserves. Our work optimizes over a substantially more complicated space of with the benchmark being Myerson's auction, which is the revenue-maximizing auction in the Bayesian setting. We do leverage several of their novel ideas, such as maintaining a differentially private internal state to guarantee approximate truthfulness.

With repeated appearances of each buyer, our auction learning problem comes to resemble dynamic mechanism design. In Bergemann and Valimaki [45] a truthful mechanism is given that exactly maximizes social welfare in a dynamic environment, and Kakade *et*

al. [46] and Pavan *et al.* [47] extended this mechanism to maximize revenue. In contrast, our mechanism approximately maximizes revenue in a dynamic environment with much looser assumptions on buyers' value distributions, but compares to the weaker per-round benchmark of the optimal single-shot revenue. A similar problem is considered in Kanoria and Nazerzadeh [48], where they consider a repeated auction with strategic bidders and complete with the single round Myerson's auction revenue but they assume that all bidders have the same value distribution and thus optimize over a simpler class of reserve price auction.

2.2.3 Differential Privacy Background

We now provide some basics on differential privacy, our main technique that helps guarantee approximate truthfulness in equilibrium. We refer to a *database* $Z \in \mathcal{Z}^n$ as a collection of data from n individuals, and we say that two databases are *neighboring* if they differ in at most one entry, i.e. they differ in one of the data points of one of the individual.

Definition 2.2.7 (Differential Privacy [49])**.** An algorithm (mapping) $\mathcal{A} : \mathcal{Z}^n \to \mathcal{R}$ is (ϵ, δ)-*differentially private* if for neighboring databases $Z, Z' \in \mathcal{Z}^n$ and subsets of possible outputs $\mathcal{S} \subseteq \mathcal{R}$, we have $P[\mathcal{A}(Z) \in \mathcal{S}] \leq \exp(\epsilon) P[\mathcal{A}(Z') \in \mathcal{S}] + \delta$. Further, if $\delta = 0$ we say that $\mathcal{A}$ is ϵ-*differentially private*.

The parameter ϵ quantifies the algorithm's privacy guarantee; smaller ϵ corresponds to stronger privacy. As a special case of Definition 2.2.7, the database may arrive online (e.g., bids in each round) and the algorithm may produce its output online (e.g., allocations and payments in each round). In this case, we still require Definition Definition 2.2.7 to hold with respect to the entire algorithmic process. See Chan *et al.* [50] and Dwork *et al.* [51] for a formal treatment of differential privacy for streams. A key property of differential privacy is that it is robust to *post-processing*.

Lemma 2.2.5 (Post-processing [49])**.** *Let $\mathcal{A} : \mathcal{Z}^n \to R$ be an (ϵ, δ)-differentially private*

algorithm and let $f : \mathcal{R} \to \mathcal{R}'$ be a random function. Then $f \circ \mathcal{A} : \mathcal{Z}^n \to \mathcal{R}'$ is also (ϵ, δ)-differentially private.

Lemma 2.2.5 is a powerful tool that says that if we compute function on the output of a differentially private algorithm, the composite function is still differentially private.

The final tool we borrow from differential privacy is the ability to maintain a histogram estimate of values which arrive one at a time. The primary technique for reporting counts involves data structures known as tree-based aggregations [51, 50]. This protocol is a differentially private method for calculating the cumulative sum of elements from 1 to t for any $t \leq T$, for which at any round t the protocol can return an estimate of the number of elements prior to round t, for which the entire execution is differentially private. We provide more details about our instances of these algorithms in Section 2.4.1.

Before we present our algorithm for repeated auctions, we first show results in Section 2.3 that show that for a single round auctions, truthful mechanisms have similar expected revenue on *similar* distributions. These tools will help us design an approximate revenue maximization auction.

2.3 Revenue Maximization on Similar Distributions

In this section, we introduce tools we willl need to argue that the revenue of our mechanism is approximately optimal. In particular, we describe how one can compare the revenue of a fixed, well-behaved mechanism on two similar product distributions $\mathbf{D}$ and $\bar{\boldsymbol{D}}$. This will imply that the problem of optimizing with respect to $\bar{\boldsymbol{D}}$ will yield approximately optimal revenue with respect to $\mathbf{D}$. These results are broadly stated and should be of independent interest.

More formally, we will consider distributions that are close in ℓ_∞ distance, and mechanisms which are well-behaved in the sense that allocating one agent leads to the exclusion of others. The relevant definitions follow. Recall that we overload the notation and D and

Definition 2.3.1 (τ-closeness)**.** We call two distributions D and $\bar{D}$ *τ-close* if $\|D - \bar{D}\|_\infty \leq \tau$.

Definition 2.3.2 (Competitiveness)**.** A truthful mechanism is *competitive* if for any valuation profile $\boldsymbol{v}$ and any pair of bidders i and j, the allocation probability $x_i(\boldsymbol{v})$ for bidder i is a non-increasing function of v_j.

In multi-unit settings, mechanisms that exactly maximize virtual surplus for any monotone virtual value function (e.g. ironed Myerson virtual value) satisfy this property. Given these definitions, we may state the main result of this subsection:

Theorem 2.3.1. *Let $\mathcal{M}$ be a competitive mechanism, and let $\mathbf{D}$ and $\bar{\boldsymbol{D}}$ be two product distributions of values such that for every bidder i, D_i and $\bar{D}_i$ are τ-close. Then the expected revenue of $\mathcal{M}$ on $\mathbf{D}$ is within an additive $2n^2h\tau$ of the revenue from $\mathcal{M}$ on $\bar{\boldsymbol{D}}$. That is, $|Rev(\mathcal{M}; \mathbf{D}) - Rev(\mathcal{M}; \bar{\boldsymbol{D}})| \leq 2n^2h\tau$.*

In fact, we prove a stronger statement than Theorem 2.3.1: we show that the revenue from each bidder is within $2nh\tau$ in each mechanism. To prove this stronger statement, we argue from the perspective an individual bidder, and consider three steps. First, we show that switching the distributions of all other bidders from $\mathbf{D}_{-i}$ to $\bar{\boldsymbol{D}}_{-i}$ does not significantly change i's allocation probability. We then show that because of the relationship between allocation and payments in truthful mechanisms an insignificant change in allocation probability implies an insignificant change in revenue. We then show that switching bidder i's distribution from D_i to $\bar{D}_i$ does not significantly impact the revenue of any mechanism. The result will follow from the triangle inequality. We begin with the first of our three steps by defining new notation for the allocation probability secured by each bidder in expectation over the other bidders.

Definition 2.3.3 (Interim allocation rule)**.** Let $\mathcal{M} = (\mathbf{x}, \mathbf{p})$ be a mechanism for the single-round game. For any value v_i, the *interim allocation rule* for i at v_i is given by $x_i(v_i) = \mathbb{E}_{\mathbf{v}_{-i}}[x_i(\mathbf{v})]$.

Our first step is to show that each bidder's interim allocation rule under competitive mechanisms is robust to small changes in other bidders' value distributions. Formally:

Lemma 2.3.2. *Let $\mathbf{D}_{-i}$ and $\bar{\boldsymbol{D}}_{-i}$ be value distributions for bidders other than i, with D_j and $\bar{D}_j$ τ-close for all $j \neq i$. Consider any truthful competitive mechanism, and let $x_i(\cdot)$ and $\tilde{x}_i(\cdot)$ denote the interim allocation rules of bidder i under $\mathbf{D}_{-i}$ and $\bar{\boldsymbol{D}}_{-i}$, respectively. Then for any value v_i, $|x_i(v_i) - \tilde{x}_i(v_i)| \leq (n-1)\tau$.*

Proof. We will consider changing the value of just one bidder, j, and observing the impact on the interim allocation rule of bidder i. The lemma will follow from repeating this argument once for each bidder other than i. To show that slightly changing bidder j's distribution has a minimal effect, we write the following sequence of equalities and inequalities, which we justify afterward.

$$
\begin{aligned}
x_i(v_i) &= \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(\mathbf{v})] D_j(v_j)\, dv_j \\
&= \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(\mathbf{v})] D_j(v_j)\Big|_0^h - \int_0^h \tfrac{d}{dv_j} \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(\mathbf{v})] D_j(v_j)\, dv_j \\
&= \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(v_i, h, \mathbf{v}_{-i,j})] - \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[\tfrac{d}{dv_j} x_i(\mathbf{v})] D_j(v_j)\, dv_j \\
&\geq \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(v_i, h, \mathbf{v}_{-i,j})] - \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[\tfrac{d}{dv_j} x_i(\mathbf{v})] (\bar{D}_j(v_j) - \tau)\, dv_j \\
&= \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(v_i, h, \mathbf{v}_{-i,j})] - \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[\tfrac{d}{dv_j} x_i(\mathbf{v})] \bar{D}_j(v_j)\, dv_j - \tau \\
&= \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(\mathbf{v})] \tilde{F}_j(v_j)\Big|_0^h - \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[\tfrac{d}{dv_j} x_i(\mathbf{v})] \bar{D}_j(v_j)\, dv_j - \tau \\
&= \int_0^h \mathbb{E}_{\mathbf{v}_{-i,j}}[x_i(\mathbf{v})] \tilde{D}_j'(v_j)\, dv_j - \tau
\end{aligned}
$$

The reasoning is as follows. The first equality is from the definition of the interim allocation rule $x_i(v_i)$. The second and third equalities follow by integration by parts and interchanging the derivative and integral, respectively. The third inequality follows from the τ-closeness of D_j and $\bar{D}_j$, and from the fact that $\frac{d}{dv_j} x_i(\mathbf{v}) \leq 0$ by the competitiveness of the mechanism. The remaining equalities follow from the same reasoning as the first three. Hence, changing bidder j's value distribution from D_j to $\bar{D}_j$ can decrease bidder i's allocation probability by at most τ. A symmetric argument bounds the increase. Further applying this same argument

to all bidders other than i, one at a time, implies the lemma. $\square$

The payment identity (Theorem 2.2.2) characterizes the payments of an individual bidder with a realized type in a truthful mechanism, and shows that this payment is completely determined by the allocation rule the agent faces. Taking expectations over the values of other agents yields a characterization of an agent's expected payments in terms of their interim allocation rule. This characterization will allow us to show that the revenue from any bidder under two similar interim allocation rules is similar.

Corollary 2.3.2.1 (of Theorem 2.2.2)**.** *In any truthful mechanism, for any bidder i with value v_i, the expected revenue of bidder i satisfies:*

$$\mathbb{E}_{\mathbf{v}_{-i}}[p_i(\mathbf{v})] = v_i x_i(v_i) - \int_0^{v_i} x_i(z)\, dz + \mathbb{E}_{\mathbf{v}_{-i}}[p_i(0, \mathbf{v}_{-i})] \tag{2.5}$$

Lemma 2.3.3. *Let x_i and $\tilde{x}_i$ be interim allocation rules for bidder i such that $|x_i(v_i) - \tilde{x}_i(v_i)| \leq \tau$ for all $v_i \in [0, h]$. If $\mathbb{E}_{\mathbf{v}_{-i}}[p_i(0, \mathbf{v}_{-i})] = 0$ under both allocation rules, then for any value v_i, the expected payments made by a bidder with that value differ by at most $2v_i\tau$ under the two allocation rules.*

Proof. By Eq. (2.5), the difference in revenue between the two mechanisms is given by

$$v_i(x_i(v_i) - \tilde{x}_i(v_i)) - \int_0^{v_i} (x_i(z) - \tilde{x}_i(z))\, dz.$$

The first term is at most $v_i\tau$. Moreover the second term is at most $\int_0^{v_i} \tau\, dz$, which is equal to $v_i\tau$. $\square$

Combining Lemma 2.3.2 and Lemma 2.3.3 yields:

Corollary 2.3.3.1. *Let $\mathbf{D}_{-i}$ and $\bar{\boldsymbol{D}}_{-i}$ be value distributions for bidders other than i, with D_j and $\bar{D}_j$ τ-close for all $j \neq i$. Consider any truthful competitive mechanism $\mathcal{M}$ where the bidders with value 0 make no payments. Then the expected revenue of $\mathcal{M}$ from bidder i differs by at most $2(n-1)h\tau$.*

We finally show that holding other bidders' value distributions fixed and switching bidder i from value distribution D_i to a τ-close distribution $\bar{D}_i$ yields similar revenue. Formally:

Lemma 2.3.4. *Let D_i and $\bar{D}_i$ be τ-close value distributions for bidder i. For any truthful mechanism $\mathcal{M}$ and any value distributions $\mathbf{D}_{-i}$ for other bidders, the expected revenue from bidder i under $D_i \times \mathbf{D}_{-i}$ and $\bar{D}_i \times \mathbf{D}_{-i}$ differ by at most $h\tau$.*

To prove Lemma 2.3.4, we use a standard characterization of a bidder's expected payments in a truthful mechanism, which can be obtained by integrating (Eq. (2.5)) over all values v_i and integrating by parts.

Corollary 2.3.4.1 (of Theorem 2.2.2)**.** *In any truthful mechanism where bidders with value 0 make no payments, for any bidder i with value distribution D_i, the expected revenue from bidder i is given by*

$$\mathbb{E}_{\mathbf{v}}[p_i(\mathbf{v})] = \int_0^h x_i'(v_i) R_i(v_i)\, dv_i \tag{2.6}$$

where $R_i(v_i)$ is bidder i's price posting revenue function, given by $R_i(v_i) = v_i(1 - D_i(v_i))$

Proof of Lemma 2.3.4. By Eq. (2.6), the difference in expected revenue between the two distributions is given by

$$\int_0^h x_i'(v_i) v_i (\bar{D}_i - D_i)\, dv_i \leq h\tau \int_0^h x_i'(v_i)\, dv_i$$

Since $\int_0^h x_i'(v_i)\, dv_i \leq 1$, the result follows. □

Now that we have all the necessarily results, we can prove the result for the main theorem in this subsection (Theorem 2.3.1)

Proof of Theorem 2.3.1. Combining Corollary 2.3.3.1 with Lemma 2.3.4 and using the triangle inequality implies that the revenue of any individual bidder i differs by at most $2(n-1)h\tau + h\tau \leq 2nh\tau$ under $\mathbf{D}$ rather than $\bar{\boldsymbol{D}}$. Summing over all bidders yields the desired bound. □

2.4 Utility-Approximate Bayesian Incentive Compatibility

In this section, we give an online algorithm (Algorithm 1) for learning the optimal auction that is approximately utility-approximate BIC. The main idea is to use differential privacy to explicitly control the amount of information the auctioneer takes forward from round t to later rounds. We do so by using differentially privacy as a tool for algorithmic stability. In Section 2.2, we learned that to run the Bayesian revenue maximizing auction, that is, the Myerson's auction, the seller needs full information about the prior $\mathbf{D}$ of the values of the bidders.

In Section 2.3, we showed that truthful mechanisms have similar revenue on bid distributions that are close in their cumulative density function values. One naive approach that the algorithm $\mathcal{A}$ could take in round t is to use the previously reported bids $\boldsymbol{b}^\tau$ for $\tau < t$ and construct empirical estimates H_i^t about the distribution D_i for each population and run a Myerson's auction based on the empirical estimate. This algorithm is prone to strategic manipulation by bidders as they can misreport their bids in earlier rounds, and make the empirical distribution converge to a distribution which leads to better outcomes for them. In other words, there are no systems in place to deter strategic behavior and the future outcomes can drastically be changing even one reported in the bid.

We overcome this challenge, by maintaining a differentially private estimate $\tilde{H}_i^t$ of each empirical bid distribution instead and choosing future auctions based only on this differentially private estimate. Since differential privacy limits how the outcomes of the algorithm change based on changing the input, from the perspective of any bidder, his behavior in the round t has very little chance of affecting any of the auctions selected in subsequent rounds. In round t, we run Myerson's mechanism with prior $\tilde{H}_i^t$ to compute allocations and payments. Thus, the one-shot mechanism in round t is exactly incentive compatible with respect to the current round.

In Algorithm 1, since we need plan to estimate the value distribution, the first step we do

is that every reported bid is rounded down to a multiple of β. Let $\mathbf{D}'$ denote the distribution obtained by rounding down $\mathbf{D}$ to nearest multiple of β. Devanur *et al.* [32] showed that the optimal revenue obtained from a rounded down distribution is only $O(\beta)$ from the optimal revenue on the original distribution. We use this fact formally later in the revenue analysis.

Thus in round t, we maintain a differentially private estimate $\tilde{H}_i^t$ of D_i but only on the rounded-down support. The algorithm then uses the Myerson's mechanism (Definition 2.2.3) defined by this differentially private estimate to decide the allocation and payment in round t. We want our estimates to be private, but also informative enough to help us run better auctions in the future. We showed in Section 2.3 that help us show that *similar* distributions lead to similar revenue in auctions, thus a *good* estimate $\tilde{H}_i^t$ of the optimal value distributions leads to *good* approximation to the optimal revenue.

Algorithm 1: Utility-Approximate BIC Online Auction

Parameters: discretization β, privacy ϵ, upper bound h, rounds T
Initialize: $\tilde{H}_i^0 \leftarrow \text{Uniform}(0, h)$ for $i = 1, \cdots, n$
for $t = 1, \cdots, T$ **do**
  Receive bid profile $\boldsymbol{v}^t = (v_1^t, \ldots, v_n^t)$, rounded down to integer multiple of β
  Run Myerson (Definition 2.2.3) with $\tilde{\boldsymbol{H}}^{t-1}$ as prior and $\boldsymbol{v}^t$ as bid for allocations/payments.
  for $i = 1, \ldots, n$ **do**
    Update $\tilde{H}_i^t$ via two-fold tree aggregation (Algorithm 2), giving as input v_i^t
  end
end

To maintain differentially private estimates, we use a standard tree-based aggregation protocol (Dwork *et al.* [51]) which we explain in detail in the following subsection.

2.4.1 Differentially Private Distribution Estimation

We now describe a differentially private procedure for estimating value distributions $\tilde{H}_i^t$ for each bidder population. This corresponds to the final operation in each round of our mechanism (Algorithm 1). Recall that the value distribution for bidder population i rounded down to the nearest multiple of is D_i', and that D_i' has finite support $\{0, \beta, 2\beta, \cdots, h\}$ of

size $h/\beta + 1$, and thus the estimates $\tilde{H}_i^t$. we maintain will have the same finite support. For now, we work under the assumption that bids are truthfully reported values, which we later validate with the incentive guarantees shortly.

Let H_i^t be the empirical (non-private) estimate of D_i' at round t. The following lemma establishes that the empirical distribution of rounded values provides a good estimate of the true distribution of rounded values, with respect to the supremum norm $\|\cdot\|_\infty$.

Lemma 2.4.1. *Let H_i^t be the empirical distribution of t i.i.d. samples from D_i'. Then, with probability at least $1-\alpha$, $\|D_i' - H_i^t\|_\infty \leq \sqrt{\frac{\log \frac{2}{\alpha}}{2t}}$.*

Proof. This is a direct result of the Dvoretzky-Kiefer-Wolfowitz inequality [52], which establishes the concentration of the empirical CDF for any distribution. □

Two-fold tree aggregation Standard tree-based aggregation [51, 50] maintains online counts of a single quantity (e.g., continuously releasing the number of 1s in a stream of bits as they come) in a differenially private manner. A naive approach to report these cumulative counts of bits so far in a (ϵ, δ)-differenitally private manner would require the algorithm to add noise to each reported cumulative sum leading to an error $O(\frac{\sqrt{T}}{\epsilon})$ over T reports. Tree-based aggregation methods can achieve the same differential privacy guarantees by adding noise to partial sums instead of the exact outputs, and can achieve the same (ϵ, δ) differential privacy guarantees with only $O(\frac{\text{polylog}(T)}{\epsilon})$ error over T reports.

The *two-fold tree aggregation* [42] to maintain an online estimate of the CDF of a probability distribution $\mathbf{D}'$ in a differentially private manner by doing the cummulative sum across two axes, the values and time. Informally, one achieves this by maintaining a private counter for each bin in a histogram over possible values which might update the counters. Since $\mathbf{D}'$ has a discrete support of $\{0, \beta, 2\beta, \cdots, h\}$, the non-private empirical CDF H_i^t can be described by a simple (increasing) step function, with steps occurring at integer multiples of β. To compute $H_i^t(u)$, the empirical CDF at value u, we need only count the number of samples from $v_i^1, \cdots, v_i^t$ which are less than u, i.e., $H_i^t(u) = (\sum_{\tau=1}^t \mathbf{1}\{v_i^\tau \leq u\})/t$.

Two-fold tree aggregation allows us to privately maintain these cumulative sums for all points $u \in \{\beta, 2\beta, \cdots, h\}$ in the support of our distributions.

The algorithm maintains n instances of the two-fold tree aggregation procedure, one for each bidder population, where each instance has its own distinct internal state. The ith instance maintains $\tilde{H}_i^t$ in round t. In each round t, the mechanism receives a value profile $\boldsymbol{v}_t$. For each population i, v_i^t is used to update the internal state of population i's tree aggregation instance.

The algorithm is given formally in Algorithm 2, and requires the following additional notation. Consider any $t \in [T]$ with binary representation $(t_{\lceil \log T \rceil}, \ldots, t_1, t_0)$. That is, $t = \sum_0^{\lceil \log T \rceil} t_j 2^j$. Let j_t be the lowest nonzero bit, and let

$$\Lambda_t = \left\{t - 2^{j_t} + 1, t - 2^{j_t} + 2, \cdots, t-1, t\right\}.$$

We also define the set,

$$\Gamma_t = \{t' : t' = t - \sum_{j=0}^{h-1} t_j 2^h, h = 1, \ldots, \lceil \log T \rceil\}.$$

We note that Γ_t has size at most $\lceil \log T \rceil$, and the set $[t]$ can be described as the union of $\lceil \log T \rceil$ such sets, i.e., $[t] = \cup_{j \in \Gamma_t} \Lambda_j$. In other words, we have constructed subsets such that

In two-fold tree aggregation, we have these aggregations over two axes: time t and value u. Thus we maintain $\frac{h}{\beta} \cdot T$ partial sums, denoted in Algorithm 2 as internal states $A_{tq} = H_i^t(u)$ for $u = q\beta$. One sample v_i^t contributes to at most $\log \frac{h}{\beta} \log T$ partial sums, and each $\tilde{H}_i^t(u)$ can be written as a sum of at most $\log \frac{h}{\beta} \log T$ partial sums.

The following lemma shows that Algorithm 2 is differentially private, and guarantees that privacy is maintained throughout the entire run of the algorithm and is a key result that we will use to prove the incentive compatibility of the algorithm.

Lemma 2.4.2 (Liu *et al.* [42])**.** *The entire stream of estimates $\{\tilde{H}_i^t\}_{t=1}^T$ output by Algorithm 2 is $\left(\epsilon, \frac{\epsilon}{T}\right)$-differentially private with respect to the input stream of bids $\{v_i^t\}_{t=1}^T$.*

Algorithm 2: Two-fold tree aggregation for population i [42]

Input: discretization parameter β, privacy parameter ϵ, upper bound on support h, number of rounds T

Internal State: Noisy partial sums A_{tq} for all $t \in [T]$ and $q \in \left[\frac{h}{\beta}\right]$

Initialize: Set $\rho = \frac{8 \log T \log \frac{h}{\beta}}{\epsilon} \sqrt{\ln \frac{T \log T \log \frac{h}{\beta}}{\epsilon}}$ and sample $A_{tq} \sim_{i.i.d.} \mathcal{N}(0, \rho^2)$ for all t and q

for $t = 1, \cdots, T$ **do**

  Receive $v_i^t = p\beta$ for some $p \in \left[\frac{h}{\beta}\right]$

  for j, k satisfying $t \in \Lambda_j$ and $p \in \Lambda_k$ **do**

    $A_{jk} = A_{jk} + 1$

  end

  for $q \in \left[\frac{h}{\beta}\right]$ **do**

    Sample $\nu_{tq} \sim \mathcal{N}(0, ((\log \frac{h}{\beta} + 1)(\log T + 1) - |\Gamma_t||\Gamma_q|)\rho^2)$

  end

  Output $\tilde{H}_i^t$, the estimated CDF: $\tilde{H}_i^t(x) := \sum_{j \in \Gamma_t} \sum_{k \in \Gamma_q} \frac{A_{jk} + \nu_{tq}}{t}$, where $q = \lfloor x/\beta \rfloor$.

end

The construction of Algorithm 2 ensures that every value $\tilde{H}_i^t(u)$ is obtained by perturbing $H_i^t(u)$ with the t-normalized sum of Gaussian variables, each with variance ρ^2. The total number of Gaussian noise terms added to obtain $\tilde{H}_i^t(u)$ is no more than $\log \frac{h}{\beta} \log T$ because the sets Γ_t and Γ_q used in the final output of Algorithm 2 have size at most $\lceil \log T \rceil$ and $\lceil \log h/\beta \rceil$, respectively. That is, for each fixed u, we have $\tilde{H}_i^t(u) - H_i^t(u) \sim N\left(0, \frac{\rho^2}{t^2} \log \frac{h}{\beta} \log T\right)$. Lemma 2.4.3 uses this fact to bound the distance between H_i^t and $\tilde{H}_i^t$.

Lemma 2.4.3. *After t rounds, for a fixed population i, with probability at least $1 - \alpha$ the empirical distribution H_i^t and the differentially private estimate $\tilde{H}_i^t$ produced by Algorithm 2 will satisfy*

$$\left\| H_i^t - \tilde{H}_i^t \right\|_\infty \leq \frac{\rho}{t} \sqrt{\log \frac{h}{\beta} \log T} \sqrt{2 \log \left(\frac{2h}{\beta\alpha}\right)},$$

for $\rho = \frac{8 \log T \log \frac{h}{\beta}}{\epsilon} \sqrt{\ln \frac{T \log T \log \frac{h}{\beta}}{\epsilon}}$.

Proof. For a Gaussian random variable $Z \sim \mathcal{N}(\mu, \rho^2)$ it holds that $P[|Z - \mu| > x\rho] \leq 2\exp(-x^2/2)$, by a standard tail bound. We apply this inequality to each point $u \in \{0, \beta, 2\beta, \cdots, h\}$ to see that, with probability at least $1 - \frac{\beta\alpha}{h}$, we have $|H_i^t(u) - \tilde{H}_i^t(u)| \leq \frac{\rho}{t}\sqrt{\log\frac{h}{\beta}\log T}\sqrt{2\log\left(\frac{2h}{\beta\alpha}\right)}$. Applying a union bound over all of the $\frac{h}{\beta}$ values of u completes the proof. □

We can now combine the previous lemmas to relate the distributions $\tilde{H}_i^t$ and D_i'.

Lemma 2.4.4. *After t rounds Algorithm 1, it holds with probability at least $1 - \alpha$ that*

$$\left\|\tilde{H}_i^t - D_i'\right\|_\infty \leq \gamma_t \quad \textit{for every } i \in [n]\,,$$

where $\gamma_t = \sqrt{\frac{\log\frac{n}{\alpha}}{2t}} + \frac{\rho}{t}\sqrt{\log\frac{h}{\beta}\log T}\sqrt{2\log\left(\frac{2hn}{\beta\alpha}\right)}$ *and* $\rho = \frac{8\log T\log\frac{h}{\beta}}{\epsilon}\sqrt{\ln\frac{T\log T\log\frac{h}{\beta}}{\epsilon}}$.

Proof. Applying the triangle inequality inLemma 2.4.3 and Lemma 2.4.1 gives the bound for a single bidder population i, and a union bound over all bidder populations n proves the lemma. □

In the remainder of this section, the same definition of γ_t will be used.

2.4.2 Incentive Guarantees for Utility-Approximate BIC Algorithm

As a corollary of Lemma 2.4.2, we can claim the following result:

Theorem 2.4.5. *The stream of estimates $\{\tilde{\boldsymbol{H}}^t\}_{t=1}^T$ maintained by Algorithm 1 and the stream of mechanism $\{\mathcal{M}^t\}_{t=1}^T$ chosen by Algorithm 1 is $(\epsilon, \epsilon/T)$-differentially private with respect to the stream of input bids $\{v^t\}_{t=1}^T$*

Proof. Algorithm 1's only record of bids which persists across rounds is its distribution estimate $\tilde{\boldsymbol{H}}^t$. In each round, it chooses an auction as a post-processing step over those estimated distributions. The two-fold tree aggregation step is $(\epsilon, \epsilon/T)$-differentially private by Lemma 2.4.2. Thus, by changing just one bid v_i^t, only the private estimate $\tilde{\boldsymbol{H}}^t$ of

that bidder is changed and thus algorithm's post-processing to estimate the virtual value distribution and select future auctions is $(\epsilon, \epsilon/T)$-differentially private by Lemma 2.2.5. $\square$

Now that we have shown that the estimates $\{\tilde{\boldsymbol{H}}^t\}_{t=1}^T$ and thus the choice of Mechanisms $\{\mathcal{M}^t\}_{t=1}^T$ is is differentially private, we can use the result to argue about incentive compatibility of the Algorithm 1. We emphasize that Theorem 2.4.5 does not claim that Algorithm 1 is itself differentially private, it only states that the procedure rests on a differentially private subroutine. This distinction is critical: our algorithm is not differentially private in its selection of allocations and payments in round t. However, the information the mechanism carries forward (namely, the estimated empirical distribution) is maintained in a differentially private manner. This is sufficient for guaranteeing that bidders' behavior in round t does not significantly affect which auctions are selected in later rounds. This will allow us to prove a utility-approximate BIC guarantee.

We note that if our mechanism were $(\epsilon, 0)$-differentially private then a result of [34], stating that any $(\epsilon, 0)$-DP mechanism is 2ϵ-dominant strategy incentive compatible.

Two issues arise if one were to try this approach in our setting. First, the entire mechanism is not differentially private as discussed above. A bidder i's behavior might have significant impact on other bidder's allocations and payments, and those bidders may as a result choose to behave differently in later rounds based on that information. Thus we relax to the weaker incentive guarantee of utility-approximate BIC, avoiding the issue of other bidders behaving differently in response to activity from earlier rounds. Second, the stream of estimates maintained by our mechanism is (ϵ, δ)-differentially private for $\delta = \epsilon/T > 0$ and not $(\epsilon, 0)$-differentially private which is necessary for the result of [34] to hold.

Theorem 2.4.6. *Algorithm 1 is $kh\epsilon\left(2 + \frac{1}{T}\right)$-utility approximate BIC when $\epsilon < 1$.*

Proof. Consider a bidder deciding how to bid in round t. They have three considerations: how their behavior will affect (1) the learning algorithm in future rounds, (2) the behavior of other bidders in future rounds, and (3) their utility in round t.

Since we seek to show that the mechanism is utility-approximate BIC, we can assume all other bidders behave truthfully in every history (by Definition 2.2.5). Therefore, other bidders will not change their behavior in future rounds, and the value of (2) is 0. The value of (3) is also 0 because the empirical Myerson auction run in each round t is chosen to be exactly truthful as a one-shot (static) mechanism, so no payer can gain anything extra in their expected utility from the current round by misreporting her bid. Thus the only utility a player can gain by lying about her value is from (1).

Next we analyze (1). Since our mechanism's differential privacy guarantee limits the extent to which a player i's report in round t affects $\tilde{H}_i^s$ for each $s > t$, and hence limits how it affects future choices of the mechanism, this allows us to control the amount of future utility she can gain from misreporting at t. Consider the change in this player's utility in all rounds $s > t \cap C_i^t$ that results from changing her bid from her true value v_i^t to any other misreport $\tilde{v_i^t}$. Let Y be the event that the bidder i bids truthfully in round t, and let $\bar{Y}$ be the event that she misreports.

Let $\text{Fut}^t(Y)$ and $\text{Fut}^t(\bar{Y})$ respectfully be the total utility the bidder achieves in all future rounds conditioned on events Y and $\bar{Y}$. That is,

$$\text{Fut}^t(Y) = E_{[\mathcal{A}, \boldsymbol{v}^\tau, \cdots, \boldsymbol{v}^T]}[\sum_{\tau > t \cap C_i^t} u_i(\mathcal{M}^\tau, \boldsymbol{v}^\tau, \boldsymbol{v}^\tau) | Y]$$

and

$$\text{Fut}^t(\bar{Y}) = E_{[\mathcal{A}, \boldsymbol{v}^\tau, \cdots, \boldsymbol{v}^T]}[\sum_{\tau > t \cap C_i^t} u_i(\mathcal{M}^\tau, \boldsymbol{v}^\tau, \boldsymbol{v}^\tau) | \bar{Y}]$$

Let $\mathcal{S}$ be the set of all possible outcomes s (that is, choice of future mechanisms) from all future rounds of the auction that this bidder may participate in. That is, outcome s is a vector of mechanisms $s = (\mathcal{M}_s^\tau)_{\tau > t+1 \cap C_i^t}$, and let $w(s)$ be the utility the bidder from outcome $s \in \mathcal{S}$, that is

$$w(s) = E_{[\boldsymbol{v}^\tau, \cdots, \boldsymbol{v}^T]}[\sum_{\tau > t \cap C_i^t} u_i(\mathcal{M}_s^\tau, \boldsymbol{v}^\tau, \boldsymbol{v}^\tau)]$$

. We can now bound the player's gain in expected utility from lying by bounding the expected value of $u(\bar{Y})$.

$$\begin{aligned}
\text{Fut}^t(\bar{Y}) &= \textstyle\int_{\mathcal{S}} w(s) P\big[s|\bar{Y}\big] ds \\
&\leq \int_{\mathcal{S}} w(s)(e^{\epsilon} P[s|Y] + \tfrac{\epsilon}{T}) ds \\
&\leq \textstyle\int_{\mathcal{S}} w(s)((1+2\epsilon) P[s|Y] + \tfrac{\epsilon}{T}) ds \\
&= (1+2\epsilon)\text{Fut}^t(Y) + \int_{\mathcal{S}} w(s) \tfrac{\epsilon}{T} ds \\
&\leq \text{Fut}^t(Y) + 2\epsilon kh + kh\tfrac{\epsilon}{T}.
\end{aligned}$$

The first inequality follows from the (ϵ, δ)-DP guarantee of Theorem 2.4.5, the second from the fact that $e^{\epsilon} \leq (1+2\epsilon)$ for $\epsilon < 1$, and the final inequality from the fact that each bidder participates in at most k rounds and her maximum utility is any round is h, so both $\text{Fut}^t(Y)$ and $\int_{\mathcal{S}} w(s) ds$ are upper bounded by kh. Thus the maximum change in utility over all future rounds between any two behaviors in the current round is therefore $2\epsilon kh + kh\frac{\epsilon}{T} = kh\epsilon(2 + \frac{1}{T})$.

Thus, the overall utility the bidder might gain from misreporting in round t is $kh\epsilon(2+\frac{1}{T})$, which converges to $2kh\epsilon$ as $T \to \infty$.

□

Now that we have proven that Algorithm 1 is $kh\epsilon\left(2 + \frac{1}{T}\right)$ utility approximate BIC, it implies that bidders can gain very little extra utility by deviating from the truthful strategy. In Section 2.3, we showed that similar distributions lead to similar revenue, using the tools we have discussed so far, we can now prove the revenue guarantee for Algorithm 1 assuming bidders don't go for this extra $kh\epsilon\left(2 + \frac{1}{T}\right)$ utility and just bid truthfully.

2.4.3 Revenue Guarantees for Utility-Approximate BIC Algorithm

Using the tools developed in Section 2.3, we can now bound the expected revenue of Algorithm 1. Since we already proved that Algorithm 1 is $kh\epsilon\left(2+\frac{1}{T}\right)$, it implies that bidders can gain very little extra utility by deviating from the truthful strategy. On the other hand, to bid optimally in each round t, the bidder i would need to know the exact priors and strategies of the other bidders, and solve an optimization problem to get the optimal utility to estimate a optimal bidding strategy. If their estimates are incorrect, then the resulting strategy could infact lead to lower utility than the one obtained by truthful bidding. Thus, if an algorithm is approximately BIC, such as Algorithm 1, it is reasonable to assume that the bidders will not try to optimize for the small extra utility of $kh\epsilon\left(2+\frac{1}{T}\right)$ and just bid truthfully.

Thus for the revenue analysis of Algorithm 1, we assume that the bidders report their true bids in each round. Recall that we use $\text{Rev}(\mathcal{M}; D)$ to denote the expected revenue generated by the mechanism $\mathcal{M}$ on a value (bid) distribution D, and that $\mathbf{D}$ and $\mathbf{D}'$ respectively denote the joint distributions of true values and true values rounded down to the nearest multiple of β. Let $\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}$, $\mathcal{M}^*_{\mathbf{D}'}$, and $\mathcal{M}^*_{\mathbf{D}}$ be the truly revenue-optimal mechanisms for the distributions $\tilde{\boldsymbol{H}}^t$, $\mathbf{D}'$, and $\mathbf{D}$, respectively. In each round of our mechanism, we get a sample from $\mathbf{D}'$ and run Myerson's auction with $\tilde{\boldsymbol{H}}^t$ as the prior; that is, we run $\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}$, hence the expected revenue of Algorithm 1 in round t is $\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}')$. Based on the regret definition given in Definition 2.2.6 of Regret, the regret of algorithm can be given by

$$\textsc{Reg}_{\mathcal{A}}(T) = T \cdot \text{Rev}(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D}) - \sum_{t=1}^{T} \text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}')$$

Here $\text{Rev}(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D})$ is the expected revenue of running a Myersons auction with the correct prior $\mathbf{D}$ which is the revenue maximizing auction in the bayesian setting.

We now present the main results of this section, Theorem 2.4.7 that bounds the average regret of Algorithm 1.

Theorem 2.4.7. *With probability at least $1-\alpha$, for regular distributions $\mathbf{D}$, value upper bound h, number of bidders n,, discretization β and privacy parameter $\epsilon < 1$, the regret of Algorithm 1 ($\mathcal{A}$) satisfies*

$$\frac{\textsc{Reg}_{\mathcal{A}}(T)}{T} \leq \beta + 4hn^2\tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{T\epsilon}\right)$$

Notice that as $T \to \infty$, the term $4hn^2\tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{T\epsilon}\right) \to 0$. Thus, if we set β, small enough such that βT is $o(T)$, we can get sub-linear regret comapred to the optimal revenue.

To prove the main revenue guarantee of Utility-Approximate BIC Online Auction presented in Algorithm 1, i.e Theorem 2.4.7, we present a few lemmas to bound the difference in revenue obtained in each round and then sum it over the T rounds to bound the average expected revenue.

In each round, Algorithm 1 runs the optimal mechanism for $\tilde{\boldsymbol{H}}^t$, but discretized value profiles are sampled from $\mathbf{D}'$. The following lemma uses results from Section 2.3 to show that the expected revenue of running $\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}$ on samples from $\tilde{\boldsymbol{H}}^t$ is not much worse that running $\mathcal{M}^*_{\mathbf{D}'}$ on samples from $\mathbf{D}'$.

Lemma 2.4.8. *With probability at least $1-\alpha$, $|Rev(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) - Rev(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}')| \leq 2hn^2\gamma_t$.*

Proof. We start by re-writing the revenue difference we wish to bound, $|\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) - \text{Rev}(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}')|$ as follows,

$$|(\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) - \text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}')) + (\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') - \text{Rev}(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}'))|.$$

For the first term inside the absolute value, Lemma 2.4.4 says that with probability at least $1-\alpha$, $\left\|D'_j - \tilde{H}^t_j\right\|_\infty \leq \gamma_t$ for all j, and therefore D'_j and $\tilde{H}^t_j$ are γ_t-close for all j. Applying Theorem 2.3.1 gives that $\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) - \text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') \leq 2hn^2\gamma_t$ with the same probability. The second term is 0 because $\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') \leq \text{Rev}(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}')$, since $\mathcal{M}^*_{\mathbf{D}'}$ is the revenue-optimal mechanism for $\mathbf{D}'$. Therefore, we get $\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) -$

$\text{Rev}(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}') \leq 2hn^2\gamma_t$.

A symmetric argument using $\mathcal{M}^*_{\mathbf{D}'}$, gives that $\text{Rev}(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}') - \text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \tilde{\boldsymbol{H}}^t) \leq 2hn^2\gamma_t$, which completes the proof. □

Now we present a result from Devanur *et al.* [32] (generalized to multiple item auction) which states that discretization of the value space by rounding down to nearest multiple of β only reduces the optimal revenue by an additive factor of β for a single item auction. Intuitively, since bids are always rounded down, this can result in a loss of at most β revenue from each of the rounds.

Lemma 2.4.9 (Devanur *et al.* [32])**.** *Rev*$(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}') \geq$ *Rev*$(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D}) - \beta$.

Combining these results, we can now bound the expected revenue of our mechanism for a fixed round.

Lemma 2.4.10. *With probability at least* $1-\alpha$*, the expected revenue obtained by Algorithm 1 in the* t^{th} *round, Rev*$(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}')$*, satisfies,*

$$\textit{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') \geq \textit{Rev}(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D}) - \beta J - 4hn^2\gamma_t,$$

for $\gamma_t = \sqrt{\frac{\log \frac{n}{\alpha}}{2t}} + \frac{\rho}{t}\sqrt{\log \frac{h}{\beta} \log T}\sqrt{2\log\left(\frac{2hn}{\beta\alpha}\right)}$ *and* $\rho = \frac{8 \log T \log \frac{h}{\beta}}{\epsilon}\sqrt{\ln \frac{T \log T \log \frac{h}{\beta}}{\epsilon}}$.

Proof. Using Lemma 2.4.8 gives,

$$\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') \geq Rev(\mathcal{M}^*_{\mathbf{D}'}; \mathbf{D}') - 4hn^2\gamma_t,$$

and applying Lemma 2.4.9 gives,

$$\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t}; \mathbf{D}') \geq Rev(\mathcal{M}^*_{\mathbf{D}}; \mathbf{D}) - \beta - 4hn^2\gamma_t.$$

□

Now that we have bounded the expected revenue in a fixed round t, we can bound the average revenue over T rounds with a union bound over all the rounds to obtain a guarantee for the average expected reveunue of our Algorithm 1 and prove the main theorem Theorem 2.4.7.

Proof of Theorem 2.4.7. We start by instantiating Lemma 2.4.10 for every round t instantiated with failure probability α/T. Then taking a union bound over all T rounds and summing over t, ensures that with probability $1-\alpha$, $\frac{1}{T}\sum_{t=1}^{T}\text{Rev}(\mathcal{M}^*_{\tilde{\boldsymbol{H}}^t};\mathbf{D}') \geq \text{Rev}(\mathcal{M}^*_{\mathbf{D}};\mathbf{D}) - \beta - \frac{4hn^2}{T}\sum_{t=1}^{T}\gamma_t$ for $\gamma_t = \sqrt{\frac{\log\frac{n}{\alpha}}{2t}} + \frac{\rho}{t}\sqrt{\log\frac{h}{\beta}\log T}\sqrt{2\log\left(\frac{2hn}{\beta\alpha}\right)}$ and $\rho = \frac{8\log T\log\frac{h}{\beta}}{\epsilon}\sqrt{\ln\frac{T\log T\log\frac{h}{\beta}}{\epsilon}}$

In the remainder of the proof, we bound $\frac{1}{T}\sum_{t=1}^{T}\gamma_t$. (Recall that the α in Lemma 2.4.10 is α/T here.)

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\gamma_t &= \frac{1}{T}\sum_{t=1}^{T}\left(\sqrt{\frac{\log\frac{nT}{\alpha}}{2t}} + \frac{\rho}{t}\sqrt{\log\frac{h}{\beta}\log T}\sqrt{2\log\left(\frac{2hnT}{\beta\alpha}\right)}\right)\\ &\leq \sqrt{\frac{2\log\frac{nT}{\alpha}}{T}} + \frac{2\rho\log T}{T}\sqrt{\log\frac{h}{\beta}\log T}\sqrt{2\log\left(\frac{2hnT}{\beta\alpha}\right)} = \tilde{O}\left(\frac{1}{\sqrt{T}} + \frac{1}{T\epsilon}\right)\end{aligned}$$

The first inequality comes from the facts that $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$ and $\sum_{t=1}^{T}\frac{1}{t} = H_T \leq \log T + 1 \leq 2\log T$. The following equality come from plugging in the expression of ρ and combining terms. □

Thus, in this chapter we showed that using tools from differential privacy and sample complexity, we can design an utility approximate BIC mechanism that leads to sub-linear revenue regret. Thus, the algorithm is able to limit the amount of strategic power the agents have over the learning procedure. In Chapter 3, we consider and alternative form of strategic behavior by the agents where instead of playing by the rules of the game and misreporting the information that was asked for them, the agent acts as an adversary and uses data corruption attacks.

CHAPTER 3

OBSERVATION-FREE ATTACKS ON STOCHASTIC BANDITS

In Chapter 2, we saw that a learner can use tools from differential privacy to prevent strategic manipulation from agents who might misreport their bids or private beliefs. In this chapter, we consider an alternate view of strategic behavior where instead of *untruthful* reporting, the agent uses unfair *data poisoning attacks* to corrupt the feedback obtained by the learner to influence future decisions of the learner towards their favor. Specifically, study data corruption attacks on stochastic multi arm bandit algorithms. We show that any bandit algorithm that makes decisions just using the empirical mean reward, and the number of times that arm has been pulled in the past can suffer from linear regret under data corruption attacks. We further show that various popular stochastic multi arm bandit algorithms such UCB, ϵ-greedy and Thompson Sampling satisfy this sufficient condition and are thus prone to data corruption attacks. We further analyse the behaviour of our attack for these algorithms and show that using only $o(T)$ corruptions, an adversarial agent can force these algorithms to select a potentially non-optimal target arm preferred by the attacker for all but $o(T)$ rounds. This chapter is based on work published in Xu *et al.* [6].

3.1 Introduction

Recall the online ad auctions example that we have been considering in Chapters 1 and 2. In Chapter 2 we discussed the case where the seller is selling the online ad impressions through repeated auction setting where strategic agents aim to misreport their values to make future decisions more favorable to them. In online advertisements, one of the main goals of the advertisers is that they want their ads to be shown and eventually clicked by as many users as possible and similarly the ad exchanges want to display ads that gets clicked more as that brings them more business and hence revenue. As a consequence, one important factor that

ad exchanges consider while deciding which ads to choose is the average *clickiness* of an ad.

Consider a case where instead of auctioning the ad impressions, the ad exchange or the learner is sequentially over T rounds is trying to decide what ad to choose from K possible advertisements based on their *clickiness*. A reasonable assumption is made that the audience for the ad impressions is stochastic, thus, conditioned that an advertisement $i \in K$ gets selected to be displayed, it gets clicked with a fixed probability ρ_i called the *click-through-rate* (*CTR*) of the advertisement. Notice that after the learner chooses an advertisement i, it only gets click feedback about the ad is actually chose. It gets no answer to question that if arm $j \neq i$ was selected, then would it have received a click? Here the partial feedback model is called bandit feedback as the learner only receives information about the arms they chose. The advertisers want their ads to be selected more often, but they are not directly involved in the decision making protocol with the learner just deciding the ads based on the click results they get from the ads. In this case, instead of using untruthful reporting, the agents can try to influence the learning process by manipulating the feedback received by the learner.

Multi-armed bandit problems provide a foundational framework for understanding sequential decision making. In the classical setting, on each round of the decision process a learner selects an action (arm,advertiser,agent) from various alternatives and, upon making this choice, receives some scalar-valued feedback/reward for the chosen action but no additional information. Algorithms for such multi-armed bandits have been widely adopted in various applications, including recommender systems [53, 54, 55, 56] and in numerous modern industry and business applications [57, 58]

As we saw in the example above, a common model assumption for bandit problems is that the reward associated with an arm is a stochastic quantity drawn from fixed distribution associated with each arm, and that this random variable is independent of the learner's previous actions. The online ad example we just described where each ads gets clicked with a fixed time independent probability, is an example of the stochastic setting called

the stochastic multi-arm bandit setting. Another model generally studied is the adversarial setting where the feedback is adversarially generated, thus leading to worst case sequences.

The stochastic model is often criticized for being unrealistic: data collected in a sequence rarely satisfy the *independent, identically-distributed (IID)* assumption, and it would be naïve to think that corruptions never occur. The advertisers want their ads to be selected more often, but they are not directly involved in the decision making protocol with the learner as the learner is just deciding the ads based on the click results they get from the ads. In this case, instead of using untruthful reporting, the agents can try to influence the learning process by manipulating the feedback received by the learner. One does not have to look hard to find pertinent examples on click fraud in online advertising [59], where the agents corrupt the a subset of the clicks observed by the decision maker by using click bots or unfair means. For example, an advertiser may hire a click farm to artificially click on some of their ads and do ghost impressions of other advertisers so that their click through rate goes up and the competitors click through rate goes down.

The adversarial model, on the other hand, is considered highly pessimistic in contexts where we expect learning to be reasonably possible. Researchers have begun to consider intermediate model assumptions, where the input data is generally assumed to be stochastic for the most part, yet a small fraction of malicious corruptions will occur, such as the click fraud example we discussed above. One more such example is fake reviews in online recommendation systems [60, 61, 62, 63].

Understanding adversarial attacks against machine learning algorithms is critical designing robust systems that can be deployed in the wild. There is a long line of work on understanding adversarial data-poisoning attacks against deep learning algorithms [64, 65, 66], supervised learning algorithms [67, 68], and more recently for multi-armed bandit problems . Perhaps the most popular algorithm for the stochastic multi-armed bandit setting, UCB [69], has a tight theoretical guarantee on its performance (i.e. its *regret*). Despite all this, it has been shown indeed that UCB is highly vulnerable to data corruption attacks

[70, 71]. In short, with only a handful of corruptions on the reward feedback given to the learning, UCB can be tricked into directing most of its choices onto a sub-optimal arm. Adversarial corruptions for multi arm bandit strategies have been studied across two axes: one line of work focus on designing and analysing different techniques to attack existing bandit algorithms [70, 71, 72, 73], while the other focuses on designing robust algorithms that can perform well under various levels of data corruption [74, 75, 76].

Notwithstanding these prior lines of work, there remains a major gap in the corruption models considered for such adversarial attacks on bandit algorithms. Most existing results assume that the adversary (corruption agent) is given full knowledge of the arm chosen by the learner and can perform a targeted corruption based on the arm selected algorithm. It has indeed been shown that all no-regret stochastic bandit algorithms are vulnerable to such powerful adversaries [71]. On the other hand, the development of robust algorithms (e.g. [74, 75]) have obtained guarantees only under a *weaker* adversary, one that can only corrupt the reward feedback *before* observing the arm selected by the learner. There has been no work, to our knowledge, that has tried to design adversarial attacks against popular stochastic bandit algorithms under the weaker adversary. For algorithms that are deterministic, which select each arm via a non-random function of prior observations, there is no relevant distinction between the strong and weak adversarial models. But given that randomization is a common and important tool in algorithm design, in this work we consider attacks against both randomized and non-randomized algorithms.

With this in mind, the goal of the present chapter is to design a strategy for adversarial attack which (a) is effective against a very broad range of multi-armed bandit algorithms and (b) fits within the weaker adversary model.

Summary of our results We show that if a stochastic bandit algorithm makes its decisions as a function of a natural statistic, the empirical mean reward and the number of pulls of each arm, then such an algorithms is fully vulnerable to the corruption attacks.

This family of bandit algorithms is indeed quite broad, and we show that most of the popular classical strategies—UCB, ϵ-greedy, and Thompson sampling [77], all of which we analyze—fall within this framework and are thus similarly vulnerable.

We further show that using by corrupting only $o(T)$ rounds, our attack can force these algorithms to select a specific arm preferred by the adversary (target arm) for all but $o(T)$ rounds. We believe this reveals what is a core flaw inherent in many bandit algorithms, and these insights can thus help to design more robust learning algorithms in this and other settings.

3.2 Preliminaries

Let's begin by formally defining the stochastic multi arm bandit setting. A principal (or learner) faces a sequential decision making problem where it needs to select one out of K actions or arms at each of the T rounds. The principal gets a reward in each round based on the arm chosen in that round. Formally, at each round t,

1. The learner decides a distribution $\pi^t \in \Delta_K$ over the K arms
2. The environment generates a reward vector $\boldsymbol{r}^t = (r_1^t, \ldots, r_K^t)$ (not observable to the principal) where $r_i^t \in [0, 1]$ is the reward the principal will receive if arm i is picked
3. The learner then selects an arm $I^t \sim \pi^t$ and receives the corresponding reward $r_{I^t}^t$ and does not observe the rest of the values in $\boldsymbol{r}^t$

For each arm i, in any round t, the reward r_i^t obtained by the learner receives for selecting an arm i is a sample from a fixed distribution such that $E[r_i^t] = \mu_i$ where μ_i is unknown to the learner. Thus by selecting the arm I^t in round t, the expected reward of the learner is μ_{I^t}

Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ be the mean reward vector that includes mean rewards of all arms. To characterize the performance of a bandit algorithm, similar to Chapter 2, a notion of regret is introduced. The regret of a bandit algorithm is defined as the gap between the

total expected reward of the algorithm and the expected reward of the algorithm that always selects the arm with the highest mean reward in each round.

Definition 3.2.1 (Regret)**.** For a bandit Algorithm $\mathcal{A}$, the regret over T rounds is defined as:

$$\textsc{Reg}_{\mathcal{A}}(T) = T \cdot \max_i \mu_i - \sum_{t=1}^{T} \mu_{I^t}$$

where I^t is the arm chosen by the algorithm in round t.

Let arm i^* be the optimal arm, i.e. $i^* = \operatorname{argmax}_i \mu_i$. Next we introduce the notion of adversarial attacks in the stochastic bandit setting. The adversarial attack is a form of data corruption where a malicious agent intends to manipulate the behavior of the bandit algorithm by corrupting the reward vector $\boldsymbol{r}^t$ generated by the environment. Specifically, the adversary can change the reward vector $\boldsymbol{r}^t$ to another corrupted reward vector $\hat{\boldsymbol{r}}^t = (\hat{r}_1^t, \ldots, \hat{r}_K^t)$ such that $\hat{r}_i^t \in [0, 1]$ for all i. We say that the round t is corrupted if the adversary changes the reward for at least one of the arms, i.e. $\|\boldsymbol{r}^t - \hat{\boldsymbol{r}}^t\|_1 > 0$. Let C be the total number of rounds that the adversary corrupts, that is $C = \sum_{t=1}^{T} \mathbb{1}\{\|\boldsymbol{r}^t - \hat{\boldsymbol{r}}^t\|_1 > 0\}$. We call C the corruption level of the bandit algorithm. Algorithm 3 gives a general framework for adversarial attacks against stochastic multi-arm bandits.

Algorithm 3: Framework for bandit learning with data poisoning attack

```
Parameters : Number of rounds T, bandit algorithm A, adversary M
for t = 1, ..., T do
    Environment generates the reward vector r^t
    if Weak attack then
        Adversary M replace the reward vector by r̂^t
    end
    Bandit algorithm A selects arm I^t
    if Strong attack then
        Adversary M observe I^t and replace the reward vector by r̂^t
    end
    Bandit algorithm A receives reward r̂^t_{I^t}
end
```

Importantly, we assume that the adversary corrupts the reward without observing the

arm selected by the adversary. That is, we restrict to the class of adversaries that perform a *Weak Attack* as defined by Algorithm 3. Formally, the protocol between the learner and the adversary at each round $t = 1, \ldots, T$ is as follows:

1. The learner decides a distribution $\pi^t \in \Delta_K$ over K arms.
2. The environment generates a stochastic reward $\boldsymbol{r}^t$.
3. The adversary corrupts the reward, and the corrupted reward becomes $\hat{\boldsymbol{r}}^t$
4. The learner picks an arm I^t from the distribution π^t and receives corrupted reward $\hat{r}^t_{I^t}$

Next we give definitions to measure the robustness of an algorithm against adversarial data corruption attacks and the power of attack methods. To characterize the performance of an algorithm under any possible adversarial attack, we introduce the definition of vulnerable algorithms.

Definition 3.2.2 (Vulnerable bandit algorithms)**.** We say a bandit algorithm $\mathcal{A}$ is vulnerable if there exists an instance and an adversary such that the adversary with $C = o(T)$ corruption level can induce linear regret $\textsc{Reg}_{\mathcal{A}}(T) = \Omega(T)$ on the bandit algorithm in expectation.

To characterize the performance of an adversarial attack, we need to consider the bandit algorithm it attempts to attack as well. The adversarial attacks that we consider in this work have a goal which is one step harder than just making the bandit algorithm obtain linear regret. The adversary has a favorite arm (that we call the target arm) and the adversary's goal is ensure that the bandit algorithm selects the target arm for most of the rounds of the algorithm. We say a bandit algorithm $\mathcal{B}$ is *completely vulnerable* to an adversarial attack $\mathcal{A}$, if with probability at least $1 - \delta(T)$, with $\delta(T) = o(1)$, the adversary can make the algorithm pick the target arm specified by the adversary for all but $o(T)$ rounds by using only $C = o(T)$ corruption level.

We now introduce a framework that is typically employed by a large class of traditional stochastic multi arm bandit algorithms. Since the goal of the bandit algorithm is to incur low

regret, to do so, it needs to figure out which arms lead to high expected rewards and then it also needs to ensure that it selects the arm with highest expected reward in most rounds. This leads to an exploration vs exploitation trade-off in the goals of the algorithm. In most cases, bandit algorithms rely on two statistics of each arm to balance the trade-off between explore and exploit: the empirical estimates on mean rewards and the corresponding variance on the estimates. The empirical means indicate which arm is likely to be the optimal, and the variances indicate how much confidence the algorithm has about its estimates. The variance of the estimate can be characterized by the number of samples the algorithm has access to for estimating the empirical means. The number of samples for each arm is exactly equal to the number of times that arm is selected by the learner in the stochastic setting. So typically, a wide class of stochastic multi arm bandit algorithms make decisions based on the empirical mean and number of selections for each arm. We call this class of algorithms as *Mean based algorithms*. Before introducing the formal definition, let us characterize the information the bandit algorithm has access to when making decisions in a round t. Let $\mathcal{I}^t$ denote the information the algorithm has access to while making decisions in round t. Using the information $\mathcal{I}^t$, the algorithm generates a probability distribution π^t over the arms where for each arm i, $\pi^t(i|\mathcal{I}^t)$ is the probability that the arm i is selected in the current round t when the information available is $\mathcal{I}^t$.

Since in each round t, the algorithm chooses an arm I^t and then obtains the corresponding reward $r^t_{I^t}$, the information obtained by the algorithm in round t is $(I^t, r^t_{I^t})$. Thus before making a decision in round t, the algorithm has access to all the information received in the rounds so far. Let us denote $\mathcal{H}^t_{\mathcal{A}} = \{(I^1, r^1_{I^1}), \ldots, (I^{t-1}, r^{t-1}_{I^{t-1}})\}$ as the history available to the algorithm $\mathcal{A}$up till round t and it is exactly the information that the bandit algorithm has access to when making the decision in this round, i.e. $\mathcal{I}^t = \mathcal{H}^t_{\mathcal{A}}$. Thus for the bandit algorithm, the decisions made in round t can be characterized by $\pi^t(i|\mathcal{I}^t) = \pi^t(i|\mathcal{H}^t_{\mathcal{A}})$.

Let $n_i^{t-1} = \sum_{\tau=1}^{t-1} \mathbb{1}\{I^\tau = i\}$ denote the number of rounds arm i gets picked by the algorithm before round t, and let $\bar{\mu}^{t-1} = \frac{\sum_{\tau=1}^{t-1} r_i^\tau \mathbb{1}\{I^\tau = i\}}{n_i^{t-1}}$ be the empirical mean of the arm i

by round t. We can define *Mean based algorithms* as follows.

Definition 3.2.3 (Mean based algorithms)**.** We say an algorithm is a mean based algorithm if

1. Its policy depends only on the empirical means $\bar{\mu}_i^{t-1}$ and number of times each arm i is selected n_i^{t-1} of all the arms. In other words for each arm i,

$$\pi^t(i|\mathcal{H}_{\mathcal{A}}^t) = \pi^t(i|n_1^{t-1}, \bar{\mu}_1^{t-1}, \ldots, n_K^{t-1}, \bar{\mu}_K^{t-1})$$

2. For each arm i, the probability that it is selected is monotonically increasing in its empirical mean, i.e.

$$\pi^t(i|\ldots, n_i^{t-1}, \bar{\mu}_i^{t-1}, \ldots) \geq \pi^t(i|\ldots, n_i^{t-1}, \bar{\mu}_i'^{t-1}, \ldots)$$

if $\bar{\mu}_i^{t-1} \geq \bar{\mu}_i'^{t-1}$

3. For each sub-optimal arm i, the probability that it is selected is monotonically decreasing on number of selections, i.e

$$\pi^t(i|\ldots, n_i^{t-1}, \bar{\mu}_i^{t-1}, \ldots) \leq \pi^t(i|\ldots, n_i'^{t-1}, \bar{\mu}_i^{t-1}, \ldots)$$

if $n_i^{t-1} \geq n_i'^{t-1}$ and $\bar{\mu}_i^{t-1} < \max \bar{\mu}_{j\in[K]}^{t-1}$.

In Definition 3.2.3, condition 1. implies that the algorithm's decisions only depends on the empirical mean and the number of pulls of each arm so far. Condition 2. implies that if the empirical mean of the arm is higher, if every other statistic remains the same, then the probability that the arm gets selected only increases. Condition 3. implies that if the arm is empirically sub-optimal, then if the number of samples used to obtain that estimate increases, then the algorithm is more confident about the fact the arm is sub-optimal, then the probability that the arm gets selected can only decrease.

Many classical bandit algorithms such as UCB, ϵ-greedy, and Thompson sampling fall

into the framework of *mean based algorithms*. In the next section we introduce our attack methodology using the adversary in consideration. Using the attack, we can show that all mean based algorithms are vulnerable to data corruptions attacks. In subsequent sections we prove stronger guarantees for a number of classical multi arm bandit algorithms by showing that UCB, ϵ-greedy, and Thompson Sampling algorithms are completely vulnerable to our attacks as long as the mean reward of the target arm is not too small.

3.2.1 Related Works

Compared to the most related works of Jun *et al.* [70], Liu and Shroff [71], and Garcelon *et al.* [73] which also study adversarial attacks against bandit algorithms, there are three fundamental differences. The first difference is that this line of work assumes that the adversary can observe the actions of the bandit algorithms, that is they consider a weak adversary (that is, strong attack as defined by Algorithm 3). This allows the adversary to attack the algorithms based on whether a particular arm is selected or not. Without such ability, to simulate their attack, the adversary need to corrupt all rounds if the bandit algorithm is randomized. The second difference is the corruption model. In their model, the corruption is counted only for the arm which is selected, while in our model, if in a round an arm is corrupted but not selected by the bandit algorithm, we still count it as a corrupted rounds. Based on our notion of corruption, the corruption budget is T for the attackers who need to corruption every round even if most of its corruption is not observed by the algorithm. The third difference is that the attacks in this line of works never apply corruptions on the target arm. Although this makes the target arm more preferred by the bandit algorithm, the negative side effect is that the amount of corruption they can apply on the non target arms is limited because they are picked less often, and thus they may have to keep attacking the algorithm. In our attack, the adversary corrupts all the arms at the beginning, making all arms look similar, thus even non-target arms are picked often enough in the early phase of the attack. This allows the adversary to apply enough corruptions on the non-target arms so

that the estimates cannot recover even after the attack stops. Through this attack, we show that all mean based algorithms which make decisions only based on estimates of empirical means are vulnerable to adversarial data corruption attacks. Liu and Shroff [71] provide a similar conclusion for the offline setting by analyzing a few specific algorithms. Also note that in the offline setting considered in Liu and Shroff [71], the algorithm receives a batch of data with size T at once, and goal of the adversary is to manipulate the algorithm's choice at the $T + 1$ round, which is very different to the online setting.

3.3 Observation-Free Attack

In this section we introduce a data poisoning attack that we call the Observation-Free Attack (Algorithm 4) which doesn't explicitly observe the behavior of the bandit algorithm while deciding how to corrupt rewards.

The attack is separated into three phases. In the first phase that lasts for C_1 rounds, the attack aims at making the algorithm receive a lot of low rewards from the optimal arm so that the empirical estimate of the optimal arm's mean reward is as low as possible and that the confidence of the algorithm over its estimate is high. To ensure that the optimal arm is picked enough times, we attack all arms which makes all arms appear equally bad to the algorithm. Explicitly, we set reward to be 0 for all arms in all the rounds in the first phase.

In the second phase that lasts for C_2 rounds, the attack tries to make the target arm distinguishable from the other arms. That is, it wants the algorithm to think that the empirical reward of the target arm is much better than all other arms. The corresponding way is to set the reward as 1 for that target arm and 0 for all other arms. Let $\tilde{i}$ be target arm, then the corrupted reward $\hat{\boldsymbol{r}}^t$ in second phase is set as $e_{\tilde{i}} \in [0, 1]^K$ where $e_{\tilde{i}}$ is the vector with 1 at the index $\tilde{i}$ and 0 everywhere else. By the end of the first two phases, the adversary has tried to ensure that empirical mean of all arms except the target arm is very low with high confidence and that the empirical mean of the target arm is much higher than the other arms.

In the third phase, the adversary does nothing and hopes that the algorithm selects the

target arm for most of the rounds and no other arm can recover from the initial corruption applied to their rewards in the first two phases. So the attack only corrupts the initial $C_1 + C_2$ rounds and the corruption level is $C_1 + C_2$.

Algorithm 4: Observation-Free Attack

```
Parameters : Number of rounds T, Mean rewards vector μ̄, bandit algorithm A,
             target arm i
Compute parameters C_1 and C_2 for the given T, μ̄, A.
for t = 1, ..., T do
    Environment generates the reward vector r^t
    if t ≤ C_1 then
        r̂^t ← (0, ..., 0)        /* Set reward as 0 for all arms */
    end
    else if C_1 < t ≤ C_1 + C_2 then
        r̂^t ← e_i^~        /* Set reward as 0 for all arms but the
         target arm.  The reward for the target arm is 1
         */
    end
    else
        r̂^t ← r^t                /* No corruption is applied */
    end
    Bandit algorithm A selects arm I^t and receives reward r̂^t_{I^t}
end
```

C_1 and C_2 are the two parameters that the adversary needs to tune based on the bandit algorithm under consideration and the rewards of the arms. For the sake of analysis, we assume that adversary has access to the mean reward for each of the arms, i.e the adversary knows $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ before the start of the bandit learning algorithm. If the adversary has access to the mean rewards, then the adversary doesn't even need to access the realized rewards from any of the rounds to decide its strategy. If the adversary does not have access to the mean rewards before the start of the process, then we show in appendix 3.7 that while corrupting the first few rounds, the adversary can observe the realized rewards to effectively estimate the mean rewards. Using the estimates, the adversary can set the parameters C_1 and C_2 of Algorithm 4 in an adaptive manner.

3.4 Vulnerability of Mean Based Bandit Algorithms

In this section we show the main result of this paper that all mean based bandit algorithms are vulnerable. In another word, any algorithm that only makes decisions that depend only on the empirical means of the arms so far and the number of time each arm has been pulled so far are not robust.

Theorem 3.4.1. *For any mean based bandit algorithm that achieves sub-linear regret in the absence of data-corruptions, there always exists an instance with an adversary data corruption attack such that the algorithm will suffer linear regret* $\textsc{Reg}_{\mathcal{A}}(T) = \Omega(T)$ *in expectation.*

To prove the theorem, we show there exist three instances such that the algorithm must suffer linear regret in at least one of the three instances. We apply observation free attack in the first instance. In the second instance, we only attack the first few rounds and show that algorithm either suffers from linear regret in this instance, or almost always picks the target arm at the second phase in the first instance. In the third instance, we apply no attack and show that either the algorithm suffers from linear regret in this instance, or only picks the optimal arm for a few rounds at the third phase in the first instance. Then if the algorithm guarantees sub-linear regret in the second and the third instance, then it must suffer from linear regret in the first instance.

Here we provide an intuition for why mean based algorithms are vulnerable. Mean based algorithms make decisions based on estimates on arms mean value and error from variance. However, the adversary could introduce additional bias to the estimates which is unknown to and omitted by the algorithms. Such bias could keep the estimates far from the real value for most of time through only small amounts of corruption, hence the algorithm will always makes poor decisions, which leads to big regret. We provide the exact proof details in Section A.1.

So far we have shown that the observation free attack can induce linear regret on the

algorithm in some instances with $\Omega(1)$ probability if such algorithm perform well in some other instances. Actually, the observation free attack is more powerful when attacking some specific mean based algorithms. In the next section we will show that UCB, ϵ-greedy, and Thompson sampling algorithms are completely vulnerable to the attack, that is, as long as the target arm has $\Omega(1)$ mean reward, the adversary with low corruption level is able to manipulate the bandit algorithm to almost always pick the target arm with high probability. Also, note that the famous EXP3 algorithm is robust in this setting as it can work even in the fully adversarial setting which includes this setting as a special case. Unlike the other classical algorithms we have just mentioned, EXP3 algorithm is not a mean-based algorithm as it doesn't use the empirical mean of rewards to make decisions.

3.5 Attack on Stochastic Bandit Algorithms

In this section we analyze the performance of the Observation-Free attack on different classical stochastic multi arm bandit algorithms including UCB, ϵ-greedy, and Thompson sampling algorithms. We show how we can tune the parameters C_1 and C_2 for each of the algorithm and present the corresponding guarantees on the vulnerability of the algorithms when subjected to our attacks.

3.5.1 Attack on UCB Algorithm

The UCB algorithm [69] is probably the most popular stochastic multi arm bandit algorithm. UCB works by maintaining upper confidence bounds on the empirical means of the arms' rewards and chooses the arm with the highest UCB value in each round. Formally, the arm selection rule of a standard UCB algorithm is the following.

$$I^t = \begin{cases} t, & \text{if } t \leq K \\ \text{argmax}_i \{\bar{\mu}_i^{t-1} + \sqrt{\frac{\log T}{n_i^{t-1}}}\}, & \text{otherwise} \end{cases} \tag{3.1}$$

where $\bar{\mu}_i^{t-1}$ and n_i^{t-1} are the empirical mean and number of times selected so far for arm

i by round t. Ties can be broke arbitrarily. Let arm i^* be the optimal arm, and arm $\tilde{i}$ be the target arm. Let $\mu = \mu_{\tilde{i}}$ denote the mean reward of the target arm for the rest of the paper.

Theorem 3.5.1. *When an adversary applies data corruption attack on UCB algorithm with the attack given by Algorithm 4, by choosing appropriate C_1 and C_2, with corruption level $C = O(\frac{K \log T}{\mu^2})$ where μ is the mean reward of the target arm, the UCB algorithm pulls the target arm for all but $O(\frac{K \log T}{\mu^2})$ rounds with probability at least $1 - 1/T$.*

The proof ideas for the analysis of attack on UCB algorithm and the other two algorithms mentioned later this section are similar. During the first stage where $t \leq C_1$, each arm will get selected for around C_1/K rounds and the empirical mean for all arms will be 0. During the second phase where $C_1 < t \leq C_1 + C_2$, the adversary starts injecting high reward for the target arm and still keeps corrupting the other arms' rewards to 0. The target arm will have the highest mean and thus will get picked most frequently. C_2 is chosen to be big enough such that the empirical mean of target arm will never be lower than its true mean with high probability. At the end of the second phase, all arms other than the target arm have been corrupted heavily. During the last stage where $t > C_1 + C_2$, since the target arm has a high enough empirical mean, it gets picked the most often. By choosing C_1 and C_2 appropriately, we can ensure that even if the other arms are explore in the third phase, they get picked so infrequently that their empirical mean cannot recover by the end of the T rounds to be better than that of the target arm. Thus, the target arm will be empirically optimal arm throughout the last phase and thus will be chosen the most often.

3.5.2 Attack on ϵ-greedy Algorithm

In ϵ-greedy Algorithm, with some probability ϵ, the algorithm decides to randomly select an arm to *explore*. Otherwise, the algorithm picks the am which is empirically best so far. Formally, the arm-selection rule of ϵ-greedy algorithm with an explore rate ϵ is:

$$I^t = \begin{cases} \text{draw uniform[K]}, & \text{w.p.}\epsilon \\ \text{argmax}_i\{\bar{\mu}_i^{t-1}\}, & \text{otherwise} \end{cases} \tag{3.2}$$

Theorem 3.5.2. *When an adversary applies data corruption attack on ϵ-greedy algorithm with the attack given by algorithm 4, by choosing appropriate C_1 and C_2, with corruption level $C = \tilde{O}(T\epsilon/\mu + K)$ where $\tilde{O}$ hides $\log T$ terms and μ is the mean reward of the target arm, the ϵ-greedy algorithm pulls the target arm for all but $\tilde{O}(T\epsilon/\mu) + K)$ rounds with probability at least $1 - \frac{2K+2}{T}$.*

For ϵ-greedy algorithm, in the absence of corruption, appropriate choice of ϵ is important to ensure sub-linear regret. The $T\epsilon$ term in unavoidable in the regret of epsilon greedy thus to ensure sub-linear regret in the absence of corruptions, the ϵ chosen by the learner has to be such that $T\epsilon$ is sub-linear. This also implies that our corruption level is also sub-linear. A typical choice is $\epsilon = O(T^{2/3})$, then the corresponding level for the attack is $\tilde{O}(T^{2/3}/\mu + K)$, and the target arm will be selected for all but $\tilde{O}(T^{2/3}/\mu + K)$ rounds with probability at least $1 - \frac{2K+2}{T}$.

3.5.3 Attack on Thompson Sampling Algorithms

Here we analyze the Thompson sampling algorithm on Bernoulli Bandits with Beta Distribution as posterior distribution [77]. In this setting, the reward from picking an arm i in any round is a Bernoulli random variable with mean μ_i. Let $\bar{\mu}_i$ be the empirical mean reward of arm i and n_i be the number of rounds when arm i is picked. At round t, for every arm i, the algorithm samples θ_i^t from the posterior distribution $B(\bar{\mu}_i \cdot n_i + 1, (1 - \bar{\mu}_i) \cdot n_i + 1)$ associated with the arm. Here $B(\cdot, \cdot)$ is a beta distribution. Then the algorithm chooses the arm with the highest sampled value, that is, $I^t = \text{argmax}_i\, \theta_i^t$.

Theorem 3.5.3. *When an adversary applies data corruption attack on the Thompson sampling algorithm with the attack given by algorithm 4, by choosing appropriate C_1 and C_2, with corruption level $C = O(\frac{2^K \log T}{\mu^2})$ where μ is the mean reward of the target arm, the*

Thompson sampling algorithm will pull the target arm for all but $O(\frac{2^K \log T}{\mu^2})$ *rounds with probability at least* $1 - \frac{2K+1}{T}$.

The theorems in this section conclude that as long as $\frac{1}{\mu^2}$ is sub linear in T where μ is the mean reward for the target arm, then an adversary using the observation free attack that ensure can the algorithms picks a target arm of their choice for all but $o(T)$ rounds with high probability. In the following section, we experimentally evaluate the performance of the difference algorithms when subjected to the observation free attack.

3.6 Experiments

In this section, to intuitively illustrate the behavior of algorithms under corruption by our adversary algorithm, we run simulations attacking UCB, ϵ-greedy and Thompson Sampling algorithm. Each algorithm is tested under the same artificial instance with 2 arms, with means $\mu_1 = 0.9$ and $\mu_2 = 0.8$. The arm 1 is the optimal arm and we set arm 2 as the target arm for the adversary. We set $T = 50000$ and the corresponding parameters (C_1, C_2) for each of the algorithm is listed in Table 3.1.

Algorithm	C_1	C_2
UCB	34	66
ϵ-greedy	150	150
Thompson Sampling	34	66

Table 3.1: Corruption level parameters for different algorithms

In Figure 3.1, we plot some key statistics about the arms as a function of the iterations that can help us understand the behaviour of the algorithms under the attacks. In Figure 3.2, we plot the number of times the optimal arm is pulled is chosen till round t, i.e. $n_{i^*}^t$ with the iteration t on the x axis in both the settings. We consider the case when there is no attack and how the number changes when we do attack the algorithm. In both Figure 3.1 and Figure 3.2, the top row zooms in on the iterations in phase 1 and 2, i.e. the corrupted rounds whereas the bottom row shows the behaviour till the horizon T.

UCB Algorithm

In UCB algorithm, the main statistic used by the algorithm is the UCB on the arms' mean reward. In each round, the arm with the highest UCB value is picked. In sub-figures $(a1)$ and $(a2)$ in Figure 3.1, we plot the UCB values for both the target arm and optimal arm. We can see sub-figures $(a1)$ that in the first phase, i.e. $t \leq C_1$ the UCB value for both the arms decreases to a value close to 0. Then in the next phase as we start injecting high rewards for the target arm, the UCB value for the target arm grows but it remains close to 0 for the optimal arm. In the third phase, after the corruption rounds, in sub-figures $(a2)$ we can see that till the end of the horizon, UCB value of the target arms remains greater than that of the optimal arm. Even the mean of the target arm decreases towards in the direction of the real mean, it never fall below the UCB of the optimal arm. In sub-figures $(a1)$ and $(a2)$ of Figure 3.2, we plot the the number of cumulative times the optimal arm gets pulled by the round t. In sub-figure $(a1)$ of Figure 3.2, we can see that in the second phase, as we start injecting higher rewards in the target arm, the algorithm completely stops choosing the optimal arm. After the second phase also, we can see in sub-figure $(a2)$ of Figure 3.2 that the optimal arm never almost never gets pulled. In the absence of corruptions, UCB algorithm performs very well and the optimal arm is pulled almost always.

ϵ-greedy Algorithm

In ϵ-greedy Algorithm, the key value to an arm's performance is its empirical mean. When there are two arms, the arm with higher empirical mean will be picked with probability $1 - \epsilon/2$. In sub-figures $(b1)$ and $(b2)$ of Figure 3.1, we plot the empirical mean for both the target arm and optimal arm. Similar to UCB we see than in Phase 1, the empirical means concentrate around 0, then the empirical mean for target arm increases in phase 2, and then the target arm remains the empirically optical arm till the end of horizon. Similar behaviour is seen in the number of times the optimal arm gets pulled. We see in sub-figures $(b1)$ and $(b2)$ of Figure 3.2 that under the attack, after Phase 1, the optimal arm gets picked very infrequently (only in explore rounds) whereas in the absence of corruptions, the optimal arm

is picked almost always.

Thompson sampling Algorithm

In Thompson Sampling algorithm, the algorithm maintains a Beta distribution for each arm. Based on the Beta distribution for the two arms, in sub-figures $(c1)$ and $(c2)$ of Figure 3.1, we plot the approximate probability that a sample from the empirical Beta distribution associated with the optimal arm is greater than a sample from the empirical Beta distribution of the target arm. Again, similar to UCB, we can see that in sub-figure $(c1)$ of Figure 3.1 that after Phase 1, the probability that the optimal arm is chosen drops close to zero. In sub-figure $(c2)$ of Figure 3.1, we observe that the optimal arm can never recover from the corruption and the probability that it gets selected remains close to 0. This is reflected in sub-figures $(c1)$ and $(c2)$ of Figure 3.2 where we can see that under attack, after phase 1, the optimal arm never gets picked whereas in the absence of corruptions, the optimal arm is picked almost always.

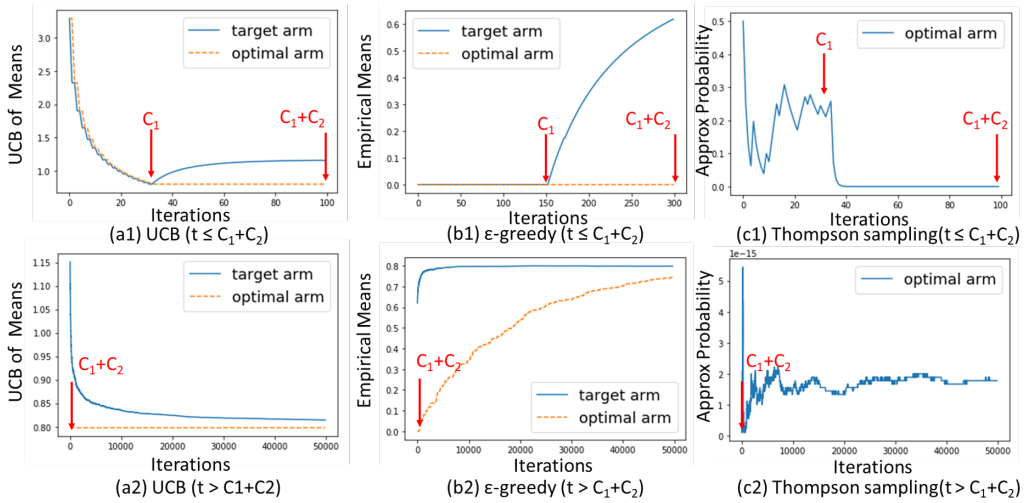


Figure 3.1: Empirical behaviors of arms in different algorithms. (a), (b) is for UCB algorithm; (c), (d) is for ϵ-greedy algorithm; (e), (f) is for Thompson sampling algorithm. (a), (c), (e) focus on the time when the rewards are being corrupted. (b), (d), (f) focus on the time when the attack stops.

To demonstrate how different algorithms behave with and without the existence of adversary, we plot the counts of the number of rounds the optimal arm gets picked versus

time in figure 3.2.

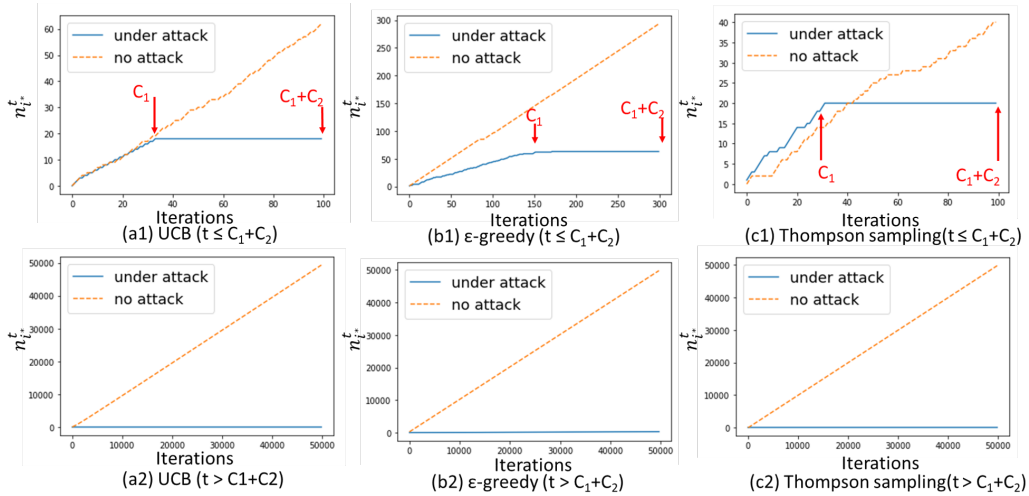


Figure 3.2: The number of rounds the optimal arm gets selected. (a1), (a2) is for UCB algorithm, (b1), (b2) is for ϵ-greedy algorithm, and (c1), (c2) is for Thompson sampling algorithm.

3.7 Attack agnostic to mean rewards of arms

We assumed in Section 3.3 that the adversary has access to mean rewards of each arm which is required to set the parameters of Algorithm 4. We can introduce a slight modification on the original attack such that the new attack can be agnostic to the mean rewards while maintaining similar performance.

The modified observation free attack works as follows. The attack is still separated into three phases and applies corruption in the same way as before. At the beginning C_1 is set to be infinite so that the attack can estimate the mean reward μ of the target arm, and once an accurate estimate is formed, the attack can set C_1 and C_2 based on the estimate. The question is how to decide the time τ when the estimating ends. Here is some intuition how we set τ. Let n^t denote the number of rounds the target arm gets selected by round t. The adversary can have a lower confidence bound on the mean reward of the target arm as $\mu_{\text{LCB}} = \bar{\mu} - \sqrt{\frac{\log T}{n_t}}$. By Hoeffding inequality, with probability at least $1 - 2/T^2$, we have $\bar{\mu} \in [\mu - \sqrt{\frac{\log T}{n_t}}, \mu + \sqrt{\frac{\log T}{n_t}}]$, which implies $\mu_{\text{LCB}} \in [\mu - 2\sqrt{\frac{\log T}{n_t}}, \mu]$. Note that

$\sqrt{\frac{\log T}{n_t}}$ diminishes from positive from infinite to 0 as n_t grows, so there exists a turning n_t^* such that $\sqrt{\frac{\log T}{n_t^*}} < \mu/4$ and $\sqrt{\frac{\log T}{n_t^*-1}} > \mu/4$. Based on this fact, the attack can stop estimating when $\mu_{\text{LCB}} \geq 2\sqrt{\frac{\log T}{n_t}}$ becomes true. At this time, with probability at least $1 - 2/T$, $\mu_{\text{LCB}} \in [\mu/2, \mu]$, in another word, $\mu_{\text{LCB}} = O(\mu)$. Then the attack can set C_1 and C_2 by setting the mean reward for the target arm as μ_{LCB}. If the time τ to set C_1 is already greater than C_1, then let $C_1 = \tau$ and determine new C_2 based on the new C_1 and μ_{LCB} correspondingly.

Lemma 3.7.1. *When attacking UCB algorithm with the new attack, with corruption level $C = O(\frac{K \log T}{\mu^2})$, the UCB algorithm will pull the target arm for all but $O(K \log T/\mu^2)$ rounds with probability at least $1-3/T$. When attacking ϵ-greedy algorithm, with corruption level $C = \tilde{O}(T\epsilon + K)$, the ϵ-greedy algorithm will pull the target arm for all but $\tilde{O}(T\epsilon + K)$ rounds with probability at least $1 - 2K + 4/T$. When attacking UCB algorithm with the new attack, with corruption level $C = O(\frac{2^K \log T}{\mu^2})$, the Thompson sampling algorithm will pull the target arm for all but $O(\frac{2^K \log T}{\mu^2})$ rounds with probability at least $1 - 3/T$.*

As we show earlier, with probability at least $1 - 2/T$, the true mean reward of the target arm satisfies $\mu \in [\bar{\mu} - \sqrt{\frac{\log T}{n_t}}, \bar{\mu} + \sqrt{\frac{\log T}{n_t}}]$. If this is true, then when the adversary determines C_1 and C_2, $\sqrt{\frac{\log T}{n_t}} \geq \mu/2$, which is equivalent to $n_t \leq \frac{4 \log T}{\mu^2}$. Note that in all the three algorithms mentioned above, n_t is at least $t/K - \sqrt{t \log T}$ with probability at least $1 - 1/T$. So the time when the adversary determine C_1 and C_2 is at most $\frac{16K \log T}{\mu^2} + K^2 \log T$, and $\mu_{LCB} \geq \mu/2$. Since $\mu_{LCB} \leq \mu$, we have $\mu_{\text{LCB}} = \Omega(\mu)$. If this C_1 is larger than the current time τ, the algorithm will stay in phase 1 until C_1 and behave exactly the same as the old attack with μ replaced by μ_{LCB}, which results in corruption level with the same order. If this C_1 is less than the current time τ, then the algorithm will set $C_1 = \tau$ instead. In this case τ is of the same order as C_1 of the old attack since $\tau = O(C_1)$ for C_1's in attacking all algorithms, the corruption level the new attack needs is still of the same order of the old one, and the probability that the new attack would fail is $2/T$ greater than the old one because of the chance that the estimation of μ is inaccurate.

CHAPTER 4

BRIDGING TRUTHFULNESS AND CORRUPTION ROBUSTNESS IN MULTI-ARM BANDIT MECHANISMS

In Chapter 2, we considered strategic manipulation by agents who report their bids untruthfully in repeated auctions. Then in Chapter 3 we considered another form of strategic manipulation where instead of strategic bidding, agents used unfair data corruption attacks to influence a bandit learning algorithm into choosing an arm of their choosing. In this chapter, we study pay-per-click ad auctions where agents bids for ad impressions but only pay if their ad impression actually gets a click. Thus, agents can employ both a) *strategic bidding* to influence the learning outcomes and b) *strategic data poisoning attacks* to corrupt the feedback obtained by the learner by manipulating the click outcomes. We show that *exploration separated* $\epsilon-$Greedy style algorithm that 1) is truthful, 2) recovers the $\tilde{O}(T^{2/3})$ lower bound in the absence of data corruptions, and 3) is robust to adversarial corruption attacks. This chapter is based on work presented in Abernethy *et al.* [7].

4.1 Introduction

Recalling the online ad auctions example we have been working with previously, consider an online ad exchange that is repeatedly selling T ad impressions over T rounds to one of K advertisers who participate in each round of the auction. In 2, we discussed how in repeated auctions, agents can misreport their bids hoping to obtain better outcomes in the future. Then in 3, we considered the setting where there is no direct bidding involved and the agent, but the agent can still use *data poisoning attacks* to manipulate a sequential learning algorithm. In this chapter, we consider a repeated pay-per-click auction with the goal of social welfare maximization over all rounds. Pay-per-click auctions as it says in the name, describe a class of auctions where once an ad is selected, the advertiser only makes a

payment to the ad exchange if the ad actually gets *clicked*.

Let us give a rough overview of the challenges of this setting when the ad platform aims to maximize *social welfare*. Social welfare is defined as the sum of utilities of the bidders and the seller. When the platform selects an ad to display to the user, the advertiser is only charged in the event that the user finds the advertisement of interest and then clicks on it. Advertisers may have a different value for clicks, thus, the social welfare generated by every click is not the same. The learner relies on the advertisers to truthfully report these values to the learner. We need to specify model parameters for these two events: assume that each ad i generates a click by a random user with probability ρ_i (the *click-through-rate*), and in the event, the ad is clicked the user makes a purchasing decision of some value. We assume that for simplicity, each agent i's value for a click is a constant μ_i across all rounds with.

One of the central challenges of designing ad auctions is that *these two parameters are not known in advance*, the principal needs to carefully manage the explore-exploit trade-off for the click statistics and design algorithms that incentivize bidders to report their value truthfully.

The agents can employ two forms of gaming to increase their individual payoff in this, potentially compromising social welfare. First, typically agents know their intrinsic value $\mu(a)$ and make a bid based on that; if misreporting this information can lead to higher individual payoff then agents will possibly do so. As a result, a line of work has focused on designing *truthful multi-armed bandit mechanisms*, i.e., making truth-telling a payoff-maximizing bidding strategy and thus rendering such attacks ineffective [78, 79].

Another attack aims to manipulate the feedback that the principal observes. For example, as shown in Chapter 3, an agent may create a bot that either clicks her own ads or does not click competing ads to adversarially bias the click-through-rate estimates and make the ad seem less desirable to the principal. A separate line of work has designed algorithms that are *robust to the presence of such adversarial corruptions* in the data [74, 75].

Notice that in absence of corruptions, if we assume all agents i report their values

μ_i truthfully, then the problem translates exactly into the stochastic multi-arm bandit we introduced in 3.2, where each action i has an expected social welfare of $\rho_i \mu_i$. Thus, Devanur and Kakade [78] and Babaioff *et al.* [79] showed that the pay-per-click auction setting can be thought of as a mechanism version of multi-arm bandits, where in addition to usual explore and exploit trade-offs the auctioneers also need to account for *strategic bidding*

4.2 Model and Preliminaries

We consider a single-slot pay-per-click (PPC) ad auction, consisting of repeated auctions with T rounds and K advertisers or *agents*. At each round, the advertisers compete for an ad impression and the auctioneer or *principal* selects one advertiser to display and a payment to charge.

Classical PPC setting. In the stochastic setting for PPC auctions, each advertiser or agent j is associated with a *click-through-rate* (CTR) ρ_j which determines the probability of getting clicked if she is selected by the principal; the click-through rates are unknown to the principal. More formally, at round t, each agent j makes a bid b_j^t and the principal displays the ad of agent I^t and charges a payment p^t which is not allowed to be above $b_{I^t}^t$. The click indicator c^t is a Bernoulli random variable with mean ρ_{I^t} that is 1 when the displayed ad gets clicked and 0 otherwise; if the click occurs, the agent pays p^t otherwise she does not pay anything (this is why the auction is called *pay-per-click*). Each agent j is also associated with a value μ_j that is fixed across rounds. We refer to this mean as mean value (or mean income). Agents are assumed to bid in a way that maximizes their expected utility which is assumed to be quasilinear, i.e., value they obtain minus payment. As introduced in Chapter 2, the welfare is defined as the total utility of all bidders and the seller. Thus, if a click happens, i.e $c^t = 1$, then the welfare of the round is equal to the value of the arm selected in round t, i.e. μ_{I^t}. Thus, by selecting an I^t, the expect welfare of the algorithm in round t is $\rho_{I^t}\mu_{I^t}$

PPC with adversarial corruptions. We extend the classical PPC setting to allow for adversarial corruptions in observed rewards. Following the model of [74] that we also considered in Chapter 3, we assume that an adversary can corrupt the results of the clicks c^t. The adversary can observe all the history of past outcomes until round t as well as the principal's distribution at round t but does not have access to the random selection of arm a^t; this is consistent to the adversarial bandit literature. The adversary has a corruption budget C which we term *corruption level* and captures the number of rounds that the adversary is allowed to corrupt. The corruption level is unknown to the principal.

Regret and performance of mechanism. The principal's performance is evaluated by *regret* which captures the loss in performance due to the principal not knowing the click-through-rates in advance. If the principal had access to the click-through-rates then the welfare-maximizing option is to select the agent j that has the highest utilization $\mu_j \rho_j$, thus the welfare regret of an algorithm $\mathcal{A}$ that chooses arm I^t in round t can be given by

Definition 4.2.1. For an algorithm $\mathcal{A}$ that chooses arm I^t in round t, the welfare regret is given by

$$\textsc{Reg}_{\mathcal{A}}(T) = T \cdot \max_j \rho_j \mu_j - \sum_{t=1}^{T} \rho_{a^t} \mu_{a^t}$$

Truthfulness requirement. In multi-armed bandit mechanisms such as pay-per-click auctions, it is important to induce the agents to not misreport their value; if the mechanism exploits the bidding pattern of an agent in order to charge them higher prices, then this creates incentive to the bidders to shade their bids which may cause the mechanism to experience unpredictable behavior. As a result, a desirable property in mechanism design is *truthfulness* which suggests that truth-telling is a dominant strategy for the agent, i.e., she cannot increase her utility by misreporting her value. The following definition quantifies this requirement for the case where the agents know their value means.

Definition 4.2.2 (Truthful Mechanism)**.** A mechanism is *truthful* if for any sets of click-through-rates and value means, click realizations, value realizations, every agent j obtains higher expected utility by bidding her true value $b_j^t = \mu_j$ at every round, irrespective of how other bidders bid; in other words, if the mechanism is truthful, misreporting does not help any agent.

The above definition means that truthful bidding is a dominant strategy for the agents when they know their value. It's a much stronger notion of truthfulness than the η-utility-approximate BIC notion we introduce in 2.2.5

A typical way to achieve truthfulness when repeatedly interacting with the same set of agents is *exploration separation*.

Definition 4.2.3 ([80])**.** For a bandit algorithm $\mathcal{A}$, we define influential rounds of $\mathcal{A}$ as the rounds from which the click realizations effects the future allocation of the algorithm. An algorithm $\mathcal{A}$ is called exploration-separated if for any click realization, all the decisions in any influential rounds are independent of the bids reported by the bidders.

Thus, for exploration-separated algorithms, the decisions in the "explore" rounds should not depend on the bids of the bidders.

The classical mechanism based on this approach for the uncorrupted case [80] is a weighted second-price auction.The mechanism first explores each arm uniformly for N rounds, then it subsequently selects the arm with the highest weight and charges a payment corresponding to the bid that would have incurred the second weight.

In the presence of adversarial corruptions, any mechanism with deterministic allocation rule is "vulnerable". Lemma 4.2.1 below specifies this fact.

Lemma 4.2.1. *For any mechanism with deterministic allocation rule, there always exists an instance and a corruption with sub-linear budget such that the mechanism will suffer linear welfare regret.*

Proof. Choose an instance (1) where there are only two arms. Their CTR' s are $\rho_1 = 0.5$, $\rho_2 = 0$, values are $\mu_1 = 1$, $\mu_2 = 1$. Let N be the expectation of the number of rounds that the second arm is selected. So the expectation of welfare regret is $0.5N$. If N is linear in T, then the mechanism suffers linear regret in this case. If N is sub-linear in T, then choose another instance (2) where $\rho'_1 = 0.5$, $\rho'_2 = 1$, $\mu'_1 = 1$, $\mu'_2 = 1$. Whenever the algorithm select arm 2, the adversary set the click result to be no click 0. Then for the mechanism, the instance is just the same as instance (1), so the second arm will be picked for N times in expectation. To make this happen, the level of corruption is N which is sub-linear, and the regret for the mechanism is $0.5(T - N)$ which is linear. □

In order to handle adversarial corruptions, we have to use mechanism with randomized allocation rules. A simple idea is to use a *truthful* version of ϵ-greedy algorithm.

4.3 Truthful corruption-robust $\epsilon-$Greedy

Inspired by the explore then commit algorithms of Devanur and Kakade [78] and Babaioff *et al.* [80], we introduce a truthful version of ϵ-Greedy inspired mechanism as 5. The algorithm sets a fixed probability ϵ and with probability ϵ, it explores by selecting a random arm uniformly and charges no price to the bidder irrespective of the click result.

In exploit rounds, the algorithm selects arm $I^t \leftarrow \mathrm{argmax}_j(\hat{\rho}^t_j \cdot b^t_j)$. In case of a click, it charges a weighted second price from arm I^t, that is, $p^t = \frac{\mathrm{smax}_j(\hat{\rho}^t_j \cdot b^t_j)}{\hat{\rho}^t_{I^t}}$. Otherwise, the payment is 0 (as is required by the PPC setting).

To make the algorithm *exploration-separated*, we ensure that the bids and allocations in explore rounds are random, and we ensure that no click results from *exploit* rounds are used for future auctions.

Algorithm 5: Truthful ϵ-greedy MAB Mechanism

Parameters : Number of arms K, Number of rounds T, exploration rate ϵ

Initialize : Set $\hat{\rho}_j^0 \leftarrow 0$, $n_j^0 \leftarrow 0$ for all $j \in [K]$

for $t = 1, \ldots, T$ **do**

  Receive bid b_j^t from arm j for all $j \in [K]$

  $\ell^t \leftarrow \begin{cases} 1 & \text{w.p. } \epsilon \\ 0 & \text{otherwise} \end{cases}$ `/* Explore or exploit */`

  if $\ell^t = 1$ **then**

    `/* ... Exploration Round ... */`

    $I^t \leftarrow j$ uniformly at random for $j \in [K]$ `/* Select arm */`

    Receive click result c^t

    Charge $p^t \leftarrow 0$ `/* Payment */`

    for $j \in [K]$ **do** `/* Update empirical mean */`

      $n_j^t \leftarrow n_{j;\ell}^{t-1} + 1_{\{j=I^t\}}$

      $\hat{\rho}_j^t \leftarrow \hat{\rho}_j^{t-1} + \dfrac{c^t \cdot 1_{\{j=I^t\}} - \hat{\rho}_j^{t-1}}{n_j^t}$

    end

  else

    `/* ... Exploitation Round ... */`

    $a^t \leftarrow \text{argmax}_j(\hat{\rho}_j^t \cdot b_j^t)$ `/* Select arm */`

    Receive click result c^t

    Charge price $p^t \leftarrow \begin{cases} \frac{\text{smax}_j(\hat{\rho}_j^t \cdot b_j^t)}{\hat{\rho}_{I^t}^t} & \textbf{if } c^t = 1 \\ 0 & \text{otherwise} \end{cases}$ `/* Payment */`

  end

end

Lemma 4.3.1. *Algorithm 5 is a truthful mechanism.*

Proof. Since bids affect only the current round, the arms do not have incentive to strategically

bid to influence future outcomes. We now analyze the effect of the bids on the current round. If the round is an exploration round, then the bid is irrelevant as the allocation is random and the payment is 0. If the round is an exploitation round, we employ a weighted second price auction which is strictly truthful for any set of weights (bids do not affect these weights). As a result, Algorithm 5 is truthful. $\square$

Welfare guarantee. For the welfare guarantee (Theorem 4.3.3), we first show that the empirical means $\hat{\rho}_j^t$ used by Algorithm 5 are always, with high probability, close the true CTR' s ρ_j for all agents j.

Lemma 4.3.2. *If the corruption level for the mechanism is C, let* $C' = C + \max\{C, 6\log(T)K/\epsilon\}$ *and* $w^t = \sqrt{\frac{2K\log(T)}{t\epsilon}} + \frac{2C'}{t}$. *With probability at least* $1 - 2(K+1)/T$, *for all* $t > 24\log(T)K/\epsilon$, $\hat{\rho}_j^t \in [\rho_j - w^t, \rho_j + w^t]$ *for all* $j \in [K]$.

Proof sketch. Denote n_j^t as the times of the jth arm get selected in explore rounds. The expectation of n_j^t is $\mathbf{E}_{n_j^t}[=]t\epsilon/K$. By Chernoff lower tail from Lemma A.2.1, taking $\delta = 1/2$, then we have $P[\{]n_j^t < \frac{t\epsilon}{2K}\} \leq 1/T^2$ when $t \geq 24K\log T/\epsilon$. Denote C_j^t as the amount of corruption that the jth arm received when it gets selected in explore rounds by round t. The expectation of C_j^t is no greater than $\frac{\epsilon C}{K}$. If $C < 6K\log T/\epsilon$, by Chernoff lower tail, taking $\delta = \frac{6K\log T}{C\epsilon}$, we have $P[\{]C_j^t > \frac{\epsilon}{K}(C + \frac{6K\log T}{\epsilon})\} \leq 1/T^2$; if $C > 6K\log T/\epsilon$, take $\delta = \sqrt{\frac{6K\log T}{C\epsilon}}$, we have $P[\{]C_j^t > \frac{2\epsilon C}{K}\} \leq 1/T^2$.

Combing above, when $t \geq 24K\log T/\epsilon$, with probability at least $1 - 2K/T$, we have $n_j^t > \frac{t\epsilon}{2K}$ and $C_j^t < (C + \max\{C, \frac{6K\log T}{\epsilon}\})\frac{K}{\epsilon}$ for all $j \in [K]$ and $t \in [T]$. Denote $C' = C + \max\{C, \frac{6K\log T}{\epsilon}\}$. Then by Hoeffding's inequality, we have $P[\{]|\hat{\rho}_j^t - \rho_j| < \sqrt{\frac{\log T}{\frac{t\epsilon}{2K}}} + \frac{2C'}{t}\} \geq 1 - 2(K+1)/T$ for all $j \in [K]$ and $t \geq 24K\log T/\epsilon \in [T]$. By setting $w^t = \sqrt{\frac{\log T}{\frac{t\epsilon}{2K}}} + \frac{2C'}{t}$, the statement of the Lemma is recovered. $\square$

Now we present the welfare guarantee for Algorithm 5.

Theorem 4.3.3. *The expected welfare regret* $\textsc{Reg}_{\mathcal{A}}(T)$ *of Algorithm 5 under corruption*

level C satisfies

$$\textsc{Reg}_{\mathcal{A}}(T) \leq T\epsilon + \sqrt{\frac{8KT\log T}{\epsilon}} + 4C'\log T + \frac{24K\log T}{\epsilon}$$

where $C' = C + \max\{C, 6\log(T)K/\epsilon\}$. *Setting* $\epsilon = T^{-1/3}(K\log T)^{1/3}$,

$$\textsc{Reg}_{\mathcal{A}}(T) \leq \widetilde{O}\left(K^{1/3}T^{2/3} + C\right)$$

where $\widetilde{O}(\cdot)$ *hides* polylog(T) *terms.*

Proof of Theorem 4.3.3. Recall that $\textsc{Reg}_{\mathcal{A}}(T) = T\rho_1\mu_1 - \sum_t \rho_{a^t}\mu_{a^t}$ where a^t is the arm picked at round t. Let's divide the $\textsc{Reg}_{\mathcal{A}}(T)$ into the regret from exploration rounds $\textsc{WelReg}_{\text{explore}}(T)$ and the regret from exploitation rounds $\textsc{WelReg}_{\text{exploit}}(T)$.

The expected number of explore rounds is $T\epsilon$. Thus $\textsc{WelReg}_{\text{explore}}(T) \leq T\epsilon$.

Define "good" event G such that at every round $t \geq 24K\log T/\epsilon$, for every arm j, $\hat{\rho}_j^t \in [\rho_j - w^t, \rho_j + w^t]$ is true. Using Lemma 4.3.2, G happens with probability at least $1 - \frac{2(K+1)}{T}$. The regret from rounds where $t < 24K\log T/\epsilon$ can be bound by $24K\log T/\epsilon$, and henceforth we only focus on the rounds $t \geq 24K\log T/\epsilon$ where event G is defined.

When G is true and $t \geq 24K\log T/\epsilon$, the regret from this round is bound by:

$$\begin{aligned} R^t &= \rho_1\mu_1 - \rho_{a^t}\mu_{a^t} \\ &\leq (\hat{\rho}_1 + w^t)\mu_1 - (\hat{\rho_{a^t}} - w^t)\mu_{a^t} \\ &\leq w^t(\mu_1 + \mu_{a^t}) \\ &\leq 2w^t \end{aligned} \tag{4.1}$$

The first inequality use Lemma 4.3.2, and the second inequality use the fact that $\hat{\rho}_1\mu_1 \leq \hat{\rho_{a^t}}\mu_{a^t}$. Summing it over t gives an upper bound on the regret from exploit rounds when G

is true and $t > 24K \log T/\epsilon$:

$$R_{\text{exploit}|G}(T) \leq \sum_{t=1}^{T} 2w^t \leq 2(\sqrt{\frac{2KT\log T}{\epsilon}} + 2C' \log T).$$

When event G doesn't occur, $R_{\text{exploit}|\overline{G}}(T)$ is at most T. We get

$$\begin{aligned}\textsc{WelReg}_{\text{exploit}}(T) &\leq 24K \log T/\epsilon + P[\{]G\} \cdot (2(\sqrt{\frac{2KT\log T}{\epsilon}}) + 2C' \log T) + P[\{]\overline{G}\}T \\ &\leq 24K \log T/\epsilon + 2(\sqrt{\frac{2KT\log T}{\epsilon}}) + 2C' \log T) + 2(K+1).\end{aligned} \tag{4.2}$$

Adding $\textsc{WelReg}_{\text{exploit}}(T)$ and $\textsc{WelReg}_{\text{exploit}}(T)$, we get

$$R(T) \leq T\epsilon + 24K \log T/\epsilon + 2\sqrt{\frac{2KT\log T}{\epsilon}} + 4C' \log T + 2(K+1).$$

Take $\epsilon = T^{-1/3}(K \log T)^{1/3}$, we have

$$\textsc{Reg}_{\mathcal{A}}(T) \leq (1+2\sqrt{2})T^{2/3}(K\log T)^{1/3} + 4C' \log T + 24K \log T^{2/3}T^{1/3} \log T + 2(K+1).$$

□

Proof. Recall that $\textsc{Reg}_{\mathcal{A}}(T) = T\rho_1\mu_1 - \sum_t \rho_{a^t}\mu_{a^t}$ where a^t is the arm picked at round t. Let's divide the $\textsc{Reg}_{\mathcal{A}}(T)$ into the regret from exploration rounds $\textsc{WelReg}_{\text{explore}}(T)$ and the regret from exploitation rounds $\textsc{WelReg}_{\text{exploit}}(T)$.

The expected number of explore rounds is $T\epsilon$. Thus $\textsc{WelReg}_{\text{explore}}(T) \leq T\epsilon$.

Define "good" event G such that at every round $t \geq 24K \log T/\epsilon$, for every arm j, $\hat{\rho}_j^t \in [\rho_j - w^t, \rho_j + w^t]$ is true. Using Lemma 4.3.2, G happens with probability at least $1 - \frac{2(K+1)}{T}$. The regret from rounds where $t < 24K \log T/\epsilon$ can be bound by $24K \log T/\epsilon$, and henceforth we only focus on the rounds $t \geq 24K \log T/\epsilon$ where event G is defined.

When G is true and $t \geq 24K \log T/\epsilon$, the regret from this round is bound by:

$$
\begin{aligned}
\textsc{WelReg}^t_{\text{exploit}} &= \rho_1\mu_1 - \rho_{a^t}\mu_{a^t} \\
&\leq (\hat{\rho_1} + w^t)\mu_1 - (\hat{\rho_{a^t}} - w^t)\mu_{a^t} \\
&\leq w^t(\mu_1 + \mu_{a^t}) \\
&\leq 2w^t
\end{aligned}
\tag{4.3}
$$

The first inequality use Lemma 4.3.2, and the second inequality use the fact that $\hat{\rho_1}\mu_1 \leq \hat{\rho_{a^t}}\mu_{a^t}$. Summing it over t gives an upper bound on the regret from exploit rounds when G is true and $t > 24K\log T/\epsilon$:

$$
\textsc{WelReg}_{\text{exploit}|G}(t) \leq \sum_{t=1}^{T} 2w^t \leq 2(\sqrt{\frac{2KT\log T}{\epsilon}} + 2C'\log T).
$$

When event G doesn't occur, $\textsc{WelReg}_{\text{exploit}|\bar{G}}(T)$ is at most T. We get

$$
\begin{aligned}
\textsc{WelReg}_{\text{exploit}}(T) &\leq 24K\log T/\epsilon + P[\{]G\} \cdot (2(\sqrt{\frac{2KT\log T}{\epsilon}}) + 2C'\log T) + P[\{]\overline{G}\}T \\
&\leq 24K\log T/\epsilon + 2(\sqrt{\frac{2KT\log T}{\epsilon}}) + 2C'\log T) + 2(K+1).
\end{aligned}
\tag{4.4}
$$

Adding $\textsc{WelReg}_{\text{exploit}|G}(T)$ and $\textsc{WelReg}_{\text{exploit}|\bar{G}}(T)$, we get

$$
\textsc{Reg}_{\mathcal{A}}(T)T\epsilon + 24K\log T/\epsilon + 2\sqrt{\frac{2KT\log T}{\epsilon}} + 4C'\log T + 2(K+1).
$$

Take $\epsilon = T^{-1/3}(K\log T)^{1/3}$, we have

$$
\textsc{WelReg}(T) \leq (1+2\sqrt{2})T^{2/3}(K\log T)^{1/3} + 4C'\log T + 24K\log T^{2/3}T^{1/3}\log T + 2(K+1).
$$

□

Note that the welfare regret degrades gracefully as the corruption level increases. It is $\widetilde{O}(T^{2/3})$ when $C = 0$ and the leading term in T remains $\widetilde{O}(T^{2/3})$ as long as $C \leq O(T^{2/3})$.

We also note that for $C > T^{2/3}$, the regret grows linear with C, and the regret bound doesn't depend on the instance.

4.4 Experiments

In this section, we present empirical analysis of the performance of our proposed algorithm 5 on synthetic data. We compare Algorithm 5, which is the truthful $\epsilon-$greedy mechanism given in this work with the explore then commit algorithm given by Babaioff *et al.* [79], and a version of Algorithm 5 that also uses the data from exploit rounds to make updates, and is thus not exploration separated or truthful.

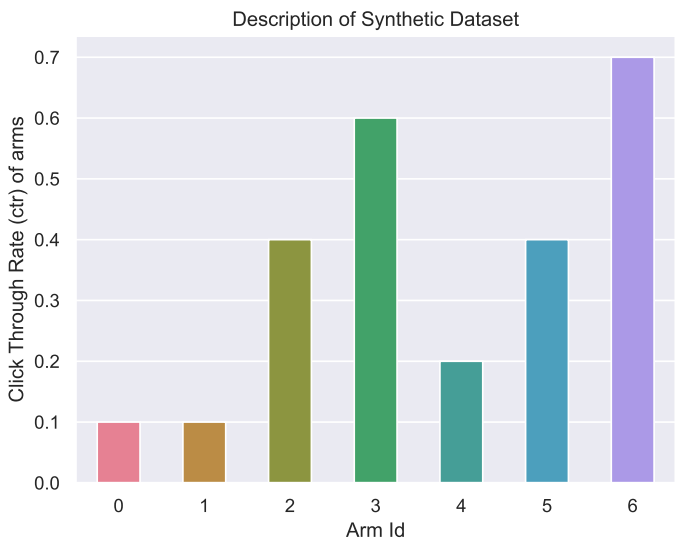


Figure 4.1: The click through rates of the arms selected for synthetic experiments

Synthetic Dataset Description We consider 7 arms with different click through rates sampled from $[0, 1]$. The exact click through rates are represented in Fig. 4.1. We set $\mu_i = 1$ for each i. We consider two instances of the problem, one where there is no adversarial corruption, and one where we use the observation free attack (4) from Chapter 3. We set the time horizon as $T = 2000$ for both the datasets and repeat each experiment 10 times. When we apply the corruption, for setting Chapter 3's parameters, we set $c1 = 150$, that is

for for the first 1000 rounds we set the reward of every arm as 0 and we set $c2 = 150$ with favorable arm 1. That is, for all arms other than arm 1, the corrupted reward is 0. For both the datasets, for all algorithms, we assume that all the bidders report their bids truthfully. Algorithm 5 and the explore then commit algorithm given by Babaioff *et al.* [79] are indeed truthful, but the non-separated version of Algorithm 5 is not.

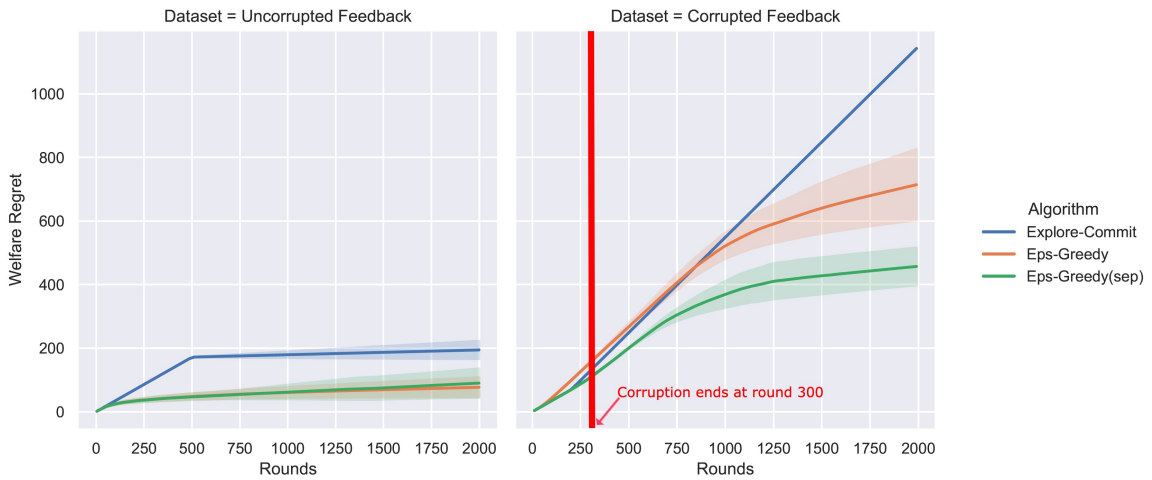


Figure 4.2: The welfare regret of Explore then Commit (Explore-Commit), Algorithm 5 (Eps-Greedy(sep), and $\epsilon-$Greedy that uses the data from explore rounds as well. The subfigure on the left represents the uncorrupted case where the subfigure on the right represents the corrupted dataset where the first 300 rounds are corrupted

Results summary The results are presented in Fig. 4.2. We can observe that under no corruptions, all three algorithms are able to converge to best arm.

In contrast, if we compare the results to the corrupted case, we can see that Explore then Commit and the $\epsilon-$Greedy that is not "exploration-separated" get highly corrupted. Even though the number of corruption rounds (300) is less than the explore round of Explore then Commit, the algorithm is never able to recover from the attack.

For Algorithm 5's version that is not exploration separated, since it uses data from exploitation rounds as well, it is still susceptible to data corruption attacks. Even though in the absence of corruption and strategic behavior, an exploration separated ϵ-Greedy algorithm has the same asymptotic guarantee as an ϵ-Greedy that uses data from exploitation rounds as well, in the presence of corruptions, the extra data turns out to be harmful for the

non exploration separated version of the algorithm. In fact, Theorem 3.5.2 from Chapter 3 shows that that $\epsilon-$Greedy algorithm that uses data in explore rounds can lead to sublinear regret.

Thus, in this case, along with ensuring truthfulness , *Exploration-separation* allowed us to be more robust to adversarial corruptions as well.

CHAPTER 5

OPTIMAL SPEND RATE ESTIMATION AND PACING FOR AD CAMPAIGNS WITH BUDGETS

Online ad platforms offer budget management tools for advertisers that aim to maximize the number of conversions given a budget constraint. As the volume of impressions, conversion rates, and prices vary over time, these budget management systems learn a spend plan (to find the optimal distribution of budget over time) and run a pacing algorithm which follows the spend plan.

This chapter considers two models for impressions and competition that varies with time: a) an episodic model which exhibits stationarity in each episode, but each episode can be arbitrarily different from the next, and b) a model where the distributions of prices and values change slowly over time. We present the first learning theoretic guarantees on both the accuracy of spend plans and the resulting end-to-end budget management system. We present four main results: 1) for the episodic setting we give sample complexity bounds for the spend rate prediction problem: given n samples from each episode, with high probability we have $|\widehat{\rho}_e - \rho_e| \leq \widetilde{O}\left(\frac{1}{n^{1/3}}\right)$ where ρ_e is the optimal spend rate for the episode, $\widehat{\rho}_e$ is the estimate from our algorithm, 2) we extend the algorithm of Balseiro and Gur [81] to operate on varying, approximate spend rates and show that the resulting combined system of optimal spend rate estimation and online pacing algorithm for episodic settings has regret that vanishes in the number of historic samples n and the number of rounds T, 3) for non-episodic but slowly-changing distributions we show that the same approach approximates the optimal bidding strategy up to a factor dependent on the rate-of-change of the distributions and 4) we provide experiments on realistic data from a large online ad platform, showing that our algorithm outperforms both static spend plans and non-pacing across a wide variety of settings. This chapter is based on results presented in Kumar *et al.* [8].

5.1 Introduction

Online advertising is a massive industry worth around $\$140$ billion dollars in 2020 in the United States alone [82]. Advertisers bidding within large online platforms are usually constrained by budget, and must decide how to distribute this budget over time as the supply and demand of impressions change. For example, there are more users online during the day than at night, leading to a variable density of impressions opportunities (see e.g. Figure 1 in Liu and Hill [83]). Furthermore, users may be more likely to interact with an ad outside of working hours, leading to those impressions generating more value for advertisers (e.g. Table 2 in Liu and Hill [83]). Finally, competition for impressions may vary over the course of the day, as other advertisers may allocate more budget to high-value periods (e.g. Figures 2 and 3 in Agarwal *et al.* [84]).

The temporal effects have led to a variety of work on constructing *spend plans* for a campaign which learn how to distribute a budget over time [85, 86, 84, 87]. Generally, the approach taken in these works is twofold: first, they use some model (e.g. a high dimensional time series model) to forecast the number of impression opportunities over the course of a day. This is taken as the spend plan. Secondly, they use a *pacing algorithm*, which tries to match the empirical spend rate to the spend plan. Lee *et al.* [87] modify the bid to control spend, while Agarwal *et al.* [84] modify their participation probability to control spend.

There are several limitations to the above approaches. First, they model the density of impression opportunities assuming that value per user and price per user is roughly constant. Considerable evidence [83, 84] refutes this assumption, suggesting that conversion rates and prices change over time. Second, their work focuses on empirical rather than theoretical results, limiting our understanding about which settings we can predict the resulting algorithms to have good performance. This motivates the problem that we study in this paper: *Can we identify non-stationary settings for which we can provably learn a spend plan that approximates the optimal distribution of budget, and where the end-to-end system*

provably performs well?

More formally: We study the problem of computing optimal spend plans from a learning-theoretic perspective in two settings: an episodic model, and a model in which price and value distributions change smoothly over time. For the first, we consider an advertiser with budget B that participates in a sequence of T single-item second-price auctions, called rounds. These auctions are divided into E episodes of $\tau = \frac{T}{E}$ rounds[1]. Each episode $e \in [E]$ has a fixed product distribution $Q_e = F_e \times D_e$, with values $v^t \sim F_e$ for $v^t \in [0, h]$ and independently prices $p^t \sim D_e$ in $p^t \in \mathbb{R}^+$. Prices and values within an episode are i.i.d., while prices and values across episodes are independently, but not identically, distributed. Let $\rho = \frac{B}{T}$ be the *average spend per round* of a strategy spending budget B over T rounds. For all e, f_e and d_e denote the probability density functions (pdf) of distributions F_e and D_e respectively. Second, we consider a non-episodic setting, where all distributions are guaranteed to change smoothly: each round has a product distribution $Q^t = F^t \times D^t$ with the property that $\|F^{t+1} - F^t\| \leq \zeta$, and $\|d^{t+1} - d^t\| \leq \theta$ for all $t \in [T]$.

For both settings, we ask: First, can we accurately estimate an optimal spend allocation? Second, given an (approximately) optimal spend plan, can we implement a pacing algorithm that satisfies the budget constraint and achieves vanishing regret compared to the ex-post optimal?

5.1.1 Main Contributions

Our main contributions are as follows.

- **Episodic Setting**. We propose a pair of algorithms 1) ApproxSpendRate, an offline algorithm that estimates the optimal spend plan on n samples, and 2) EpisodicAdaptivePacing, an online algorithm that adaptively follows the spend plan over T new auctions, that jointly have regret vanishing in n and T, compared to the best bidding

[1]We assume equal sized episodes to simplify the presentation of the paper. Our results can be generalized to different sized episodes where the size of episodes can also be estimated.

strategy in hindsight. The formal statement appears as Theorem 5.4.4 in Section 5.4 and relies on the following additional results:

- **Estimating Optimal Spend Plan.** In Section 5.3 we bound the accuracy of constructing of an optimal spend plan. We give an algorithm ApproxSpendRate, that given n samples from each episode, with probability at least $1-\frac{2E}{n}$, produces a spend plan that satisfies $|\widehat{\rho}_e - \rho_e| \leq (E+1) \cdot \widetilde{O}\left(\frac{1}{n^{1/3}}\right)$ where ρ_e is the optimal spend rate for the episode, $\widehat{\rho}_e$ is the estimate from our algorithm and E is the number of episodes.
- **Online Pacing Algorithm on Spend Plan.** In Section 5.4 we then give an adaptive pacing algorithm EpisodicAdaptivePacing that takes an (approximately accurate) spend plan, and implements a bidding strategy that follows this spend plan over T new auctions. The regret of this algorithm vanishes in n and T with respect to the best bidding strategy in hindsight.

- **Slow-moving Distributions**. In Section 5.5, for slow-moving distributions we learn a spend plan as if the data came from an episodic model with number of episodes E. The end-to-end performance achieves a constant factor approximation of the to the best bidding strategy in hindsight, where the constant factor depends on the rate at which the distributions change.

- **Experiments on Realistic Data**. Finally, in Section 5.6 we present experiments on realistic data from a large online advertising platform. We compare the performance of our method to the Balseiro and Gur [88] algorithm (which neither estimates nor uses a spend plan as it was designed for adversarial and stationary settings). Our method compares favorably to the ex-post optimal strategy and outperforms other methods in a wide variety of settings.

5.1.2 Related Work

Optimal Spend Rate Estimation. There are number of works that aim to estimate optimal spend rates for budget pacing [85, 86, 84, 87]. Ma *et al.* [85] and Agarwal *et al.* [86] primarily focus on the on the spend plan estimation. Both of these papers aim to forecast user visits, which correlates strongly with the number of impression opportunities. They do this using time series modeling techniques for users within the targeting criteria of a campaign. These works do not attempt to estimate how conversion rates or prices for ad opportunities change over time. Lee *et al.* [87] and Agarwal *et al.* [84] combine user visit estimates with an online pacing algorithm to match the spend rate to the user visit rate. Similar to the approaches below, Lee *et al.* [87] uses a multiplicative shading strategy (i.e. bidding $\alpha \cdot v^t$ instead of v^t) to control spend, while Agarwal *et al.* [84] participate in each auction with a parameterized probability to control spend. None of these papers give formal guarantees on the performance of the end-to-end budget management system.

Online Algorithms for Pacing. Work on pacing algorithms has only focused on guarantees for pacing algorithms in absence of a spend plan. In many cases, for repeated second-price auctions, the optimal pacing strategy in hindsight is a multiplicative shading strategy (i.e. bidding $\alpha \cdot v^t$ for the auction at time t for a fixed $\alpha \leq 1$ that does not vary over time) [89, 90, 91, 81, 92]. Balseiro and Gur [81] were the first to give online learning algorithms that approximate this best response. For i.i.d. value and price distributions, they give an online algorithm with regret $O(T^{1/2})$. Similar guarantees are also shown by Balseiro *et al.* [93] who achieve $O(T^{1/2})$ regret for stationary value and price distributions setting without assuming independence between values and prices. There are few works that give provable guarantees for non-stationary competition and values. Balseiro and Gur [81] consider the case of adversarial values and prices and show that no algorithm can achieve sub-linear regret with respect to any benchmark that obtains more than $\frac{B}{Th}$ fraction of the utility obtained by using the optimal strategy with the power of hindsight (where h is an upper bound on the value). They also give an algorithm which obtains the $\tilde{O}(T^{1/2})$

upper bound on the regret with respect to $\frac{B}{Th}$ fraction of the optimal. Balseiro *et al.* [93] considers both an ergodic setting and a periodic setting where regret grows as $\tilde{O}(T^{1/2})$. Their algorithms do not construct a spend plan and instead rely on the fact that at a macro-level the expected optimal spend rate is constant. By contrast, in our setting obtaining no-regret may depend on saving enough budget for the end of the campaign (for example to reach users on the weekend for a week-long campaign). Only by explicitly constructing an approximately optimal spend plan can one give guarantees for such campaigns.

Conitzer *et al.* [94] show that for individual first-price single-item auctions, multiplicative shading yields the Eisenberg-Gale outcome of the corresponding Fisher market (though generally multiplicative shading is not a best response in this setting). Gao *et al.* [95] give an online learning algorithm for this setting that results in this equilibrium and can be run in a decentralized way by each advertiser individually.

While bid modification yields the an optimal strategy for a bidder, an alternative way to respect a budget constraint is to limit the number of auctions a bidder participates in. Mehta *et al.* [96] give revenue guarantees for the online matching problem where users (in this case, impressions for sale) arrive one at a time and the auction selects a winner who pays their bid; once a bidder has exhausted their budget they will no longer be selected as a winner. Subsequently bidder selection has been applied to more general settings [97, 98, 99, 100]. Since truthful bidding is not a best response for advertisers in bidder selection mechanisms, this line of work is less directly relevant to our work.

In the previous two lines of work, advertisers know the value they have for an impression when they bid. A separate line of work considers a bandit setting, where the value is only revealed to advertisers after they win an auction. Amin *et al.* [101] and Tran-Thanh *et al.* [102] give theoretical guarantees for discrete value distributions. Flajolet and Jaillet [103] extend these results to continuous distributions. Finally, Nuara *et al.* [104] and Avadhanula *et al.* [105] consider the problem of allocating budget across different channels. The different channels have different distributions and as such bear some similarity to the setting we

consider. However, since all channels are simultaneous available and each channel is i.i.d., the spend rate remains constant over time (cf. our setting where spend rates change).

Equilibrium Analysis In addition to the work on online algorithms, there's a growing body of work that analyzes the equilibria of pacing systems under the assumption that all advertisers use the same bid-shading approach, e.g. [106, 107, 108, 92, 94, 109, 110]. The framework of Balseiro *et al.* [111] studies stationary equilibria and characterize Bayesian optimal mechanisms that satisfy budget constraints.

5.2 Setting and Preliminaries

We study the problem of designing a bidding algorithm for budget-constrained advertisers in non-stationary settings. This bidding algorithm aims to maximize utility subject to a given budget constraint B. The algorithm participates in a sequence of T single-item second-price auctions[2]. We refer to each auction as a round. In the following we present the notation for the episodic setting that we study, the non-episodic setting is formally introduced in Section 5.5.

In every round $t \in [T]$, the bidder observes a value v^t for the impression opportunity[3] and submits a bid b^t to the auctioneer. Let p^t be the highest competing bid for the impression opportunity. When $b^t \geq p^t$ the bidder wins, spends p^t, and gains utility $u^t = v^t - p^t$. Otherwise she loses, pays 0, and gains utility $u^t = 0$. The bidder's goal is to maximize their utility subject to the sum of expenditures across T rounds being at most B.

A strategy σ of the bidder is a sequence of deterministic[4] mappings $\sigma^1, \ldots, \sigma^t$ where σ^t uses the information that is available to bidder in round t to produce bid $b^t_{\sigma^t}$. We focus

[2]Our model captures additional settings, including posted prices and second-price auctions with reserve prices, but for ease of exposition we consider second-price auctions throughout.

[3]v^t could capture the value v that the advertiser has for a conversion times the probability of a conversion of the impression opportunity. The latter may depend on context like the user or a search query and is estimated by the platform.

[4]While all of our algorithms are deterministic, the lower bound in Lemma 5.2.1 can be extended to randomized mappings as well. For ease of exposition, our algorithms use only deterministic strategies.

attention on strategies that respect the budget constraint.

Definition 5.2.1 (Budget-feasibly Strategy)**.** σ is budget feasible if $\sum_{t=1}^{T} \mathbb{1}\left\{b_{\sigma^t}^t \geq p^t\right\} p^t \leq B$ for any realized values $\boldsymbol{v} = v^1, \ldots, v^t$ and prices $\boldsymbol{p} = p^1, \ldots, p^t$.

A strategy's utility is simply its total utility over T rounds.

Definition 5.2.2 (Performance of a Strategy)**.** For a given budget feasible strategy σ, it's performance on a realized sequence of values $\boldsymbol{v}$ and prices $\boldsymbol{p}$ is given by

$$\pi^{\sigma}(\boldsymbol{v}; \boldsymbol{p}) = \sum_{t=1}^{T} \mathbb{1}\left\{b_{\sigma^t}^t > p^t\right\}(v^t - p^t). \tag{5.1}$$

As benchmark, we consider the best (fractional) allocation in hindsight on the realized values $\boldsymbol{v}$ and prices $\boldsymbol{p}$. While the benchmark may appear to be strong, it is commonly used in the budget pacing literature.

Definition 5.2.3 (Hindsight Strategy Benchmark)**.** The performance of the hindsight strategy H on a realized sequence of values $\boldsymbol{v}$ and prices $\boldsymbol{p}$ is given by

$$\pi^{H}(\boldsymbol{v}; \boldsymbol{p}) = \max_{x \in [0,1]^T} \sum_{t=1}^{T} (v^t - p^t) x^t \quad \text{s.t.} \quad \sum_{t=1}^{T} p^t x^t \leq B$$

Here $x^t \in [0, 1]$ represents a fractional allocation of impression opportunities. We measure the regret of a strategy compared to the benchmark in expectation over the values and prices. We use the notion of α-regret proposed by [112], a multiplicative notion:

Definition 5.2.4 (α-regret)**.** For strategy σ and $\alpha \in (0, 1]$, the α-regret with respect to the hindsight strategy is:

$$\alpha\text{-}\textsc{Reg}_{\sigma}(T) = \alpha \mathbf{E}\left[\pi^{H}(\boldsymbol{v}, \boldsymbol{p})\right] - \mathbf{E}[\pi^{\sigma}(\boldsymbol{v}, \boldsymbol{p})]$$

Where the expectation is over $(\boldsymbol{v}, \boldsymbol{p})$ sampled from $\vec{Q}$, that is, (v^t, p^t) in episode e is sampled from $Q_e = F_e \times D_e$.Our algorithm first constructs a *spend plan* prior to the T

auctions, using historical data. The accuracy of the spend plan will be a function of the sample size our algorithm is given.

Definition 5.2.5 (Sample Complexity)**.** The sample complexity of achieving a given approximation factor $1 - \epsilon$ is the minimum number of samples m such that there exists an (offline) learning algorithm A with the desired approximation.

Of particular interest are algorithms where both the α-regret is sublinear in T, and additionally, α approaches 1 using a polynomial number of samples. We overload the term "vanishing regret" for such situations.

Definition 5.2.6 (Vanishing Regret)**.** A strategy σ_n (which has access to n samples from $\vec{Q}$) has *vanishing regret* if $(1-\epsilon)$-$\textsc{Reg}_{\sigma_m}(T) = o(T)$ *and* $m \in O(\text{poly}(\epsilon^{-1}))$.

5.2.1 Outline of the Solution

As mentioned previously, our algorithm first produces a spend plan from data, then uses a pacing algorithm to meet that spend plan. The former is an offline learning problem that happens before the campaign starts. The latter is an online algorithm that operates on the spend plan and realized expenditures. Before going into these components, it is informative to understand why this decomposition in a spend plan and pacing algorithm makes sense.

Why Historical Data is Needed. Balseiro *et al.* [111] have studied pacing for non-stationary distributions without using historical data. Could the episodic setting that we're studying be amenable to positive results without historical data too? Unfortunately this is not the case. The following is an example with two episodes for deterministic algorithms. We generalize the example in the lemma that follows.

Example 5.2.1. Consider two instances of the episodic setting characterized by the episodic distributions, $I = (Q_1, Q_2)$ and $I' = (Q_1, Q_2')$ (where $Q = F \times D$). All the distributions consist of a single atom: prices distributions $D_1 = D_2 = D_2'$ and yield 1 with probability 1. The value generated by F_1 is 2, by Q_2 is 1 and by Q_2' is 3.

Consider a buyer with budget $B = \frac{1}{2}T$, thus they can buy precisely half the impression opportunities. In both instances, the first episode yields utility $2 - 1 = 1$ per round that is won, but the second episode differs for the two instances. For I, the per-round utility when the bidder wins is $1 - 1 = 0$, while for I' it is $3 - 1 = 2$.

So if the bidder faces the first instance, she needs to win all but a sublinear (in T) number of rounds in episode 1 for vanishing regret, but if she faces the second instance she may win at most a sublinear number of rounds in the first episode. Since she doesn't know which instance she faces until she enters episode 2, any strategy must incur $\Omega(T)$ regret on at least one of I, I'.

The example above can be generalized to a stronger result for instances with more episodes and that includes randomized algorithms.

Lemma 5.2.1. *Any strategy σ that only depends on the history $\mathcal{H}^t = (v^i, b^i, p^i)_{i=0}^{t-1} \cup v^t$ in round t, for a large enough T, and budget B such that $0 < \rho = \frac{B}{T} < h$, for any number of episodes E, for any ϵ such that $1 - \epsilon > \max\{\frac{\rho}{h}, \frac{1}{E}\}$, there exists an instance of the episodic setting with distributions $\vec{Q} = (Q_1, \cdots, Q_E)$ such that*

$$(1 - \epsilon)\text{-}\textsc{Reg}_\sigma(T) \geq \Omega(T).$$

Proof. In an adversarial setting where the values and prices are arbitrary, in each round t, Balseiro and Gur [81, Theorem 1] show that for any strategy σ such that σ^t depends only on the history $\mathcal{H}^t$, for a large enough T, for any budget B satisfying $\rho = \frac{B}{T} < h$ where h is the upper bound on the values, for any ϵ such that $(1 - \epsilon) > \frac{\rho}{h}$, there exists adversarial values $\boldsymbol{v}$ and $\boldsymbol{p}$ such that $(1 - \epsilon)\pi^H(\boldsymbol{v}; \boldsymbol{p}) - \pi^\sigma(\boldsymbol{v}; \boldsymbol{p}) \geq \Omega(T)$.

Note that if $\rho \geq h$ then truthful bidding is feasible and achieves the optimal utility. When $\rho < h$, then the above result says that there exists a barrier of $\frac{\rho}{h}$ such that for any strategy σ that only depends on the history, there always exists an instance where σ cannot obtain better than $\frac{\rho}{h}$ fraction of the optimal utility. In other words, if the budget constraint is active,

the lower the budget, the smaller the fraction of the optimal budget constrained utility the advertiser can hope to attain.

Since we don't make any assumptions about how distributions Q_e are related across the episodes, we can extend the analysis and the adversarial case example of Balseiro and Gur [81, Theorem 1] to work in the episodic setting and show that similar lower bounds can be shown for such strategies in our setting as well. Specifically, in the proof of Balseiro and Gur [81, Theorem 1], the adversarial example has the value v^t fixed as h for all T rounds, and the price profile is samples from a distribution such that the T round auction is divided into m *episodes*. When $\frac{h}{E} \leq \rho = \frac{B}{T} < h$, then by setting $m = E$ in the proof for Balseiro and Gur [81, Theorem 1], we can recover the guarantee that there exists an instance such that ϵ-$\textsc{Reg}_\sigma(T) \geq \Omega(T)$ for $1 - \epsilon \geq \frac{\rho}{h}$. If $0 < \rho < \frac{h}{E}$, then the complete example is scaled down by replacing h with $h' = \rho E$. Note that since $\rho < \frac{h}{E}$, we have that $h' < h$, thus the example is valid. We also get that $\frac{h'}{E} \leq \rho = \frac{B}{T} < h'$. Thus, we obtain that ϵ-$\textsc{Reg}_\sigma(T) \geq \Omega(T)$ for $1 - \epsilon \geq \frac{\rho}{h'} = \frac{1}{E}$. □

Since algorithms in the episodic setting that only operate on the immediate history fail to have vanishing regret, we use access to historical data in the form of samples from the distribution.

Why Spend Plans are Needed. With access to samples from the distribution, one could still attempt to design an algorithm that does not involve a spend plan. Recall from the related work that ex-post the optimal bidding strategy is to bid $\beta^* \cdot v^t$ for some constant β^*. So what if we used samples to estimate this β and used this directly? The following lemma shows that this yields linear regret with constant probability.

Lemma 5.2.2. *There exists an instance of the episodic setting with distributions $\vec{Q} = (Q_1, Q_2)$ such that the ex-ante optimal pacing multiplier β^* incurs $O(T)$.*

Proof. Consider a two episode setting with $\tau = T/2$ rounds in each episode and total budget as $B = T/2$. Let the price of each item in both the episodes be 1. For the first episode, let

the value of each item be 2 and in the second episode, the item has value 4 with probability $1/2$ and value 0 with probability $1/2$.

The ex-post optimal strategy strategy, or the hindsight strategy buys all $v = 4$ items in the second episode, and half of the 2 value items in the first episode, for expected value $\frac{T}{2} \cdot 4 \cdot \frac{1}{2} + \frac{T}{2} \cdot 2 \cdot \frac{1}{2} = \frac{3T}{2}$, i.e. a utility of $\frac{3T}{2} - \frac{T}{2} = T$. Any fixed shading pacing strategy must either win all or none of the items in the first episode. In the former case, the strategy has no budget for episode 2, resulting in total utility of $\frac{T}{2} \cdot 2 - \frac{T}{2} = \frac{T}{2}$. When the pacing strategy loses all first episode impressions, enough budget remains to win all the 4 value items in second episode, for expected utility of $\frac{T}{2} \cdot 4 \cdot \frac{1}{2} - \frac{T}{2} = \frac{T}{2}$.

So, any fixed shading parameter earns expected utility at most $\frac{T}{2}$. Thus, the regret of any fixed multiplicative pacing algorithm will be $T - \frac{T}{2} = \frac{T}{2} = \Omega(T)$. □

Instead we construct an intermediate spend plan and combine this with an adaptive pacing algorithm; an approach we outline next.

Outline of the Solution using Spend Plans. To understand the optimal spend plan, we first introduce the notation of spend functions which represent the expected expenditure of strategies that shade by a fixed shading multiplier.

Definition 5.2.7 (Spend Function)**.** Consider a fixed pacing strategy σ_μ that always bids $b^t = \frac{1}{\mu+1} \cdot v^t$ and is not restricted by any budget constraints. The expected expenditure of σ_μ in a single round in episode e is

$$\overline{G_e}(\mu) = \mathbf{E}_{v \sim F_e, p \sim D_e}[\mathbb{1}\{v \geq (1+\mu)p\}\, p] \tag{5.2}$$

$$= \int_0^h (1 - F_e((1+\mu)p)) \cdot p \cdot d_e(p) dp. \tag{5.3}$$

Definition 5.2.8 (Optimal Spend Rates)**.** Given an episodic setting where the ex-post optimal bidding strategy is to bid $\beta^* \cdot v^t$, we define $\boldsymbol{\rho} = \rho_1, \cdots, \rho_E$ as optimal spend rates if for all e, $\rho_e = \overline{G_e}(\mu^\star)$, where $\frac{1}{1+\mu^\star} = \beta^*$.

In simpler words, the optimal spend plan is characterized by the optimal spend rates $\boldsymbol{\rho} = \rho_1, \cdots, \rho_E$, such that the ρ_e is equal to expected expenditure of the ex post optimal bidding strategy in a single round in episode e. We define the dual of the expectation of the optimization problem in Definition 5.2.3 as

$$\Psi(\mu) = \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\sum_{t=1}^{T}\left(v^t - (1+\mu)p^t\right)^+ + \mu B\right]. \tag{5.4}$$

The ex-post optimal bidding strategy $\sigma_{\mu^\star}$ bids $\frac{v^t}{1+\mu^\star}$ where $\mu^\star$ is the dual minimizer ($\Psi(\mu^\star) = \inf_{\mu \geq 0} \Psi(\mu)$), which spends the complete budget in expectation:

$$\tau \sum_{e=1}^{E} \overline{G_e}(\mu^\star) = \tau \sum_{e=1}^{E} \rho_e = B, \tag{5.5}$$

where $\rho_1, \cdots, \rho_E$ are the optimal spend rates. Refer to Appendix B.1 for a detailed exposition about the characterization of the optimal spend rates through the dual of the problem. Knowing the optimal spend rates can help decompose the entire campaign into smaller budget constrained campaigns for each episode where the distributions of values and prices remain stationary. The exact formulation of optimal spend rates requires complete knowledge of the distributions $\vec{Q} = (Q_1, \cdots, Q_E)$ which is not available, instead we have access to historic samples from the distribution $\vec{Q}$. This is reasonable to assume as we are designing this framework for large online ad exchanges which usually have access to a lot of historical data. Our solution is is a two step pipeline:

1. **Approximate optimal spend rates:** Use historical samples from $\vec{Q}$ to approximate optimal spend rates $\boldsymbol{\rho} = \rho_1, \cdots, \rho_E$ as $\widehat{\boldsymbol{\rho}} = \widehat{\rho}_1, \cdots, \widehat{\rho}_E$.

2. **Adaptive pacing with spend rates:** Use the approximate spend rates to construct an online pacing algorithm that runs on realized impressions.

5.2.2 Preliminaries

We will use some results on uniform convergence and pacing for i.i.d. settings in this paper.

Dvoretzky-Kiefer-Wolfowitz (DKW) Inequality

The Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [113, 52] gives a uniform convergence bound on the empirical cumulative distribution function.

Lemma 5.2.3 (DKW Inequality)**.** *Given n samples $X_1, X_2, \ldots, X_n$ from a distribution F. The empirical cdf on the samples is given by $\widehat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_i < x\}$. With probability at least $1-\delta$*

$$\left\| F - \widehat{F} \right\| \leq \sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$

Kernel Density Estimation

While the DKW inequality 5.2.3 gives strong uniform convergence bounds on the cdf of a distribution, bounding the probability density function (pdf) of a distribution is more challenging. A common approach to do this is to use Kernel Density Estimation (KDE) [114, 115]. Let D be the distribution with pdf d that we want to estimate as $\widehat{d}$. Formally the Kernel Density Estimation is defined below for scalar distributions which we use in our setting.

Definition 5.2.9 (Kernel Density Estimation)**.** Given a kernel K, scalar s, and n samples $X_1, \cdots, X_n$ from the distribution D, the KDE is given by

$$\widehat{d}(x) = \frac{1}{n \cdot s} \sum_{i=1}^{n} K\left(\frac{x - X_i}{s}\right).$$

While most results on KDE are for bounding the mean squared error ($\mathbf{E}_d\left[(\widehat{d} - d)^2\right]$), recent work by Jiang [116] gives a uniform convergence guarantee for KDE. We present a simplified version of their result below.

Lemma 5.2.4 ([116]). *If d is Lipschitz and bounded i.e. there exists a constant C_1 such that $|d(x) - d(x')| \leq C_1|x - x'|$ for $x, x' \in \mathbb{R}$ and $\|d\| \leq C_2$ for some constant C_2, then there exists a constant C' (that depends on K, C_1, C_2, and some other constants), such that with probability at least $1 - 1/n$ and by setting $s = n^{-1/3}$, the kernel density estimate $\widehat{d}$ satisfies that*

$$\left\|\widehat{d} - d\right\| \leq C' \left(s + \sqrt{\tfrac{\log n}{ns}}\right) = \widetilde{O}\left(\frac{1}{n^{1/3}}\right).$$

provided that K is spherically symmetric, non-increasing, and has exponential decay (i.e. $K(x) = k(|x|)$ where $k : \mathbb{R}^+ \to \mathbb{R}^+$ is a non decreasing function s.t. for all $u > u_\eta$, $k(u) \leq C_\eta \exp(-u^\eta)$ for some fixed η, C_η, and u_η)

A number of popular kernel choices such that Gaussian, exponential, uniform, and many more satisfy the requirements of Lemma 5.2.4. While the Lipschitz requirement appears strong, a large number of common distributions such as the normal distribution, Cauchy distribution, exponential distributions and lognormal distributions have Lipschitz and bounded pdfs.

Balseiro-Gur Pacing Algorithm

Consider a setting with just one episode such that the values and prices in every round t are sampled from fixed stationary distributions F and D. Balseiro and Gur [81] give an adaptive pacing algorithm based on minimizing the dual $\Psi(\mu)$. In every round t, the algorithm bids $\frac{v^t}{1+\mu^t}$ and the pacing parameter μ^t is updated using a projected gradient decent style update in the direction that minimizes the dual.

Lemma 5.2.5 ([81]). *If the value and prices in each round are samples from a stationary distribution such that $\Psi(\mu)$ is thrice differentiable in μ with bounded gradients and is strongly convex, then using the Adaptive Pacing algorithm from Balseiro and Gur [81] with $\eta = O(T^{-1/2})$ results in strategy $\mathcal{A}$ with*

$$\mathbf{E}\left[\pi^H(\boldsymbol{v},\boldsymbol{p})\right] - \mathbf{E}[\pi^\sigma(\boldsymbol{v};\boldsymbol{p})] \leq O(\sqrt{T}).$$

5.3 Approximating Optimal Spend Rates

We first turn our attention to estimating optimal spend plans in the episodic setting. Given n samples from $\vec{Q}$, we will divide our budget B across E episodes by estimating target spend rates $(\widehat{\rho}_1, \ldots, \widehat{\rho}_E)$ that approximate the optimal spend rates $(\rho_1, \ldots, \rho_E)$ additively. The main theorem we'll prove in this section is the following.

Theorem 5.3.1. *Given B, T, E, sample oracles F_e and D_e, where d_e is Lipschitz and bounded, sampling budget n, K, setting $s = O(n^{-1/3})$, w.p. $\geq 1 - \frac{2E}{n}$, for each episode e, ApproxSpendRate returns $\widehat{\rho}_e$ s.t.*

$$|\widehat{\rho}_e - \rho_e| \leq (E+1) \cdot \widetilde{O}\left(\frac{1}{n^{1/3}}\right).$$

ApproxSpendRate (Algorithm 6) is based on the fact that the ex-post optimal bidding strategy spends the complete budget in expectation. The resulting algorithm consists of three main steps: i) use historical samples to approximate spend functions $\overline{G_e}(\mu)$ for each episode as $\widehat{G_e}(\mu)$, ii) use Eq. (5.5) to approximate $\mu^\star$ as $\widehat{\mu}$, and iii) estimate the expected spend per round for each episode using the approximate spend functions and $\widehat{\mu}$.

Algorithm 6: ApproxSpendRate

Input: Budget B, rounds T, episodes E, sampling oracles F_e, D_e,Kernel K, scalar s

for $e = 1, \ldots, E$ **do**

  Samples n values $\vec{V} = (V_1, V_2, \ldots, V_n) \sim F_e$

  Samples n prices $\vec{P} = (P_1, P_2, \ldots, P_n) \sim D_e$

  $\widehat{G_e}(\mu) \leftarrow$ ApproxSpendSP$(n, \vec{V}, \vec{P}, K, s)$

end

$\widehat{G}(\mu) \leftarrow \frac{1}{E}\sum_{e=1}^{E} \widehat{G_e}(\mu)$

$\widehat{\mu} = \min \mu$ s.t. $\widehat{G}(\mu) \leq \frac{B}{T}$

return $(\widehat{\rho}_1, \ldots, \widehat{\rho}_E)$ where $\forall e$, $\widehat{\rho}_e \leftarrow \widehat{G_e}(\widehat{\mu})$

Before we discuss how to approximate the spend functions $\overline{G_e}(\mu)$, in Lemma 5.3.2 we show that a good approximation of $\overline{G_e}(\mu)$ allows for a good approximation of the optimal spend rates $\rho_1, \ldots, \rho_E$.

Lemma 5.3.2. *In Algorithm 6, for each episode e if the estimated episodic spend function $\widehat{G_e}$ obtained the end of Line 15 satisfies $\left\|\widehat{G_e} - \overline{G_e}\right\| \leq \gamma$, then for each e, the algorithm returns spend rate $\widehat{\rho}_e$ such that $|\widehat{\rho}_e - \rho_e| \leq (E+1)\gamma$.*

Proof. The proof progresses in two steps: First, we show that the episodic guarantee in the premise of the lemma yields a bound for the overall spend function. Next, we show that the approximate spend functions when evaluated on $\widehat{\mu^\star}$ yield provable bounds on the resulting episodic spend rates, where $\widehat{\mu^\star}$ is the optimal pacing parameter learned using the estimated overall spend function.

First note that the episodic bounds yield a bound on the overall spend function $\widehat{G}()$.

$$
\begin{aligned}
|\widehat{G}(\mu) - \overline{G}(\mu)| &= |\frac{1}{E}\sum_{1}^{E}\widehat{G_e}(\mu) - \frac{1}{E}\sum_{1}^{E}\overline{G_e}(\mu)| && (5.6)\\
&= \frac{1}{E}|\sum_{1}^{E}(\widehat{G_e}(\mu) - \overline{G_e}(\mu))| \\
&\leq \frac{1}{E}\sum_{1}^{E}|\widehat{G_e}(\mu) - \overline{G_e}(\mu)| && (5.7)\\
&\leq \frac{1}{E}\cdot E\cdot\gamma && (5.8)\\
&= \gamma && (5.9)
\end{aligned}
$$

Where Eq. (5.6) follows from the definition of $\widehat{G}(\mu)$ and Eq. (B.6), Eq. (5.7) follow from the triangle inequality, and Eq. (5.8) follows from the fact that $\left\|\widehat{G_e} - \overline{G_e}\right\| \leq \gamma$. Since this holds for arbitrary μ, this implies that $\left\|\widehat{G} - \overline{G}\right\| \leq \gamma$.

Using Eq. (B.7), and the formulation of the algorithm, we know that $\widehat{G}(\widehat{\mu}) = \overline{G}(\mu^\star) = \frac{B}{T}$, and Eq. (5.9) implies $|\widehat{G}(\widehat{\mu}) - \overline{G}(\widehat{\mu})| \leq \gamma$. Combining the two, we get $|\overline{G}(\widehat{\mu}) - \overline{G}(\mu^\star)| \leq \gamma$

It can easily be shown that for any episode e, $\overline{G_e}(\mu) = \mathbf{E}_{(v,p)\sim Q_e}[\mathbb{1}\{v \geq (1+\mu)p\}\, p]$ is

a monotonically decreasing function in μ. Consider $\mu_1 \leq \mu_2$, since all values v and prices p are non-negative, for any v and p, $\mathbb{1}\{v \geq (1+\mu_2)p\}\, p \leq \mathbb{1}\{v \geq (1+\mu_1)p\}\, p$.

So there are two possible case, 1) $\mu^\star < \widehat{\mu}$ or $\mu^\star \geq \widehat{\mu}$.

Consider Case 1) i.e. $\mu^\star < \widehat{\mu}$, then

$$
\begin{aligned}
|\overline{G}(\widehat{\mu}) - \overline{G}(\mu^\star)| &= \overline{G}(\mu^\star) - \overline{G}(\widehat{\mu}) && (5.10)\\
&= \frac{1}{E}\sum_{e=1}^{E}(\overline{G_e}(\mu^\star) - \overline{G_e}(\widehat{\mu}))\\
&\geq \frac{1}{E}\max_e(\overline{G_e}(\mu^\star) - \overline{G_e}(\widehat{\mu})) && (5.11)
\end{aligned}
$$

Where Eq. (5.10) and Eq. (5.11) follow from the monotonicity of $\overline{G}$ and $\overline{G_e}$. Similarly, using the other direction for the case $\mu^\star \geq \widehat{\mu}$, we get that for every e,

$$|\overline{G_e}(\widehat{\mu}) - \overline{G_e}(\mu^\star)| \leq E \cdot |\overline{G}(\widehat{\mu}) - \overline{G}(\mu^\star)| \leq E \cdot \gamma$$

Now consider $|\widehat{\rho}_e - \rho_e|$ for some e,

$$
\begin{aligned}
|\widehat{\rho}_e - \rho_e| &= |\widehat{G_e}(\widehat{\mu}) - \overline{G_e}(\mu^\star)|\\
&= |\widehat{G_e}(\widehat{\mu}) - \overline{G_e}(\widehat{\mu}) + \overline{G_e}(\widehat{\mu}) - \overline{G_e}(\mu^\star)|\\
&\leq |\widehat{G_e}(\widehat{\mu}) - \overline{G_e}(\widehat{\mu})| + |\overline{G_e}(\widehat{\mu}) - \overline{G_e}(\mu^\star)|\\
&\leq \gamma + E \cdot \gamma
\end{aligned}
$$

Thus, for all e, it holds that $|\widehat{\rho}_e - \rho_e| \leq (E+1)\gamma$. □

5.3.1 Approximating spend functions

Recall from Definition 5.2.7 that for an episode e with value distribution F_e and price distribution D_e,

$$\overline{G_e}(\mu) = \int_0^h (1 - F_e((1+\mu)p)) \cdot p \cdot d_e(p)\, \mathrm{d}p.$$

This implies that if we can approximate F_e and d_e, we can use Eq. (5.3) to approximate $\overline{G_e}(\mu)$. In Algorithm 7, we use the empirical estimate $\widehat{F_e}$ of F_e, and use Kernel Density Estimation to approximate d_e as $\widehat{d_e}$.

Algorithm 7: ApproxSpendSP: Stochastic prices.

Input:$(V_1, \ldots, V_n)$: values samples, $(P_1, \ldots, P_n)$: price samples, Kernel function K, scalar s

$\widehat{d}(p) \leftarrow \frac{1}{n \cdot s} \sum_{i=1}^{n} K\left(\frac{p - P_i}{s}\right)$

$\widehat{F}(v) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{V_i < v\}$

$\widehat{G}(\mu) \leftarrow \int_0^h p(1 - \widehat{F}((1+\mu)p))\widehat{d}(p)$

return $\widehat{G}(\mu)$

The estimate of the spend function satisfies the following uniform convergence bound.

Lemma 5.3.3. *Given n samples from F_e and D_e (where d_e is Lipschitz and bounded), setting $s = O(n^{-1/3})$ ApproxSpendSP (Algorithm 7) returns the approximate episodic spend function $\widehat{G_e}$ such that with probability at least $1 - 2/n$ it holds that $\left\|\widehat{G_e} - \overline{G_e}\right\| \leq \widetilde{O}\left(\frac{1}{n^{1/3}}\right)$.*

Proof. Consider any $\mu \geq 0$, with probability at least $1 - \frac{2E}{n}$,

$$
\begin{aligned}
&|\widehat{G_e}(\mu) - \overline{G_e}(\mu)| \\
&= |\int_0^h (1 - F_e((1+\mu)p)) \cdot p \cdot d_e(p)\, \mathrm{d}p - \int_0^h (1 - \widehat{F_e}((1+\mu)p)) \cdot p \cdot \widehat{d}(p)\, \mathrm{d}p| \\
&\leq \int_0^h |(1 - F_e((1+\mu)p)) \cdot d_e(p) - (1 - \widehat{F_e}((1+\mu)p))\widehat{d_e}(p)| \cdot p\, \mathrm{d}p \\
&\leq \int_0^h \left(|d_e(p) - \widehat{d_e}(p)| + |\widehat{F_e}((1+\mu)p)\widehat{d_e}(p) - F_e((1+\mu)p)d_e(p)|\right) \cdot p\, \mathrm{d}p \\
&\leq \int_0^h \left(\widetilde{O}\left(\frac{1}{n^{1/3}}\right) + \widetilde{O}\left(\frac{1}{n^{1/3}}\right) + \widetilde{O}\left(\frac{1}{n^{1/2}}\right) + \widetilde{O}\left(\frac{1}{n^{5/6}}\right)\right) \cdot p\, \mathrm{d}p \\
&= h \cdot \widetilde{O}\left(\frac{1}{n^{1/3}}\right) \\
&= \widetilde{O}\left(\frac{1}{n^{1/3}}\right)
\end{aligned}
$$

where the first and second inequality follow from triangle inequality. For the third step, we use the PDF and CDF concentration bounds, and the fact for any $0 \leq a, b, c, d \leq 1$,

$|ab - cd| \leq |c - a| + |d - b| + |(c - a).(d - b)|$.

Here $\widetilde{O}$ notation hides the polylog(n) terms along with constants like h, C' from Lemma 5.2.4 and $\|d_e\|$. $\square$

Combining the results of results of Lemma 5.3.2 and Lemma 5.3.3 completes the proof of Theorem 5.3.1. Theorem 5.3.1 implies that using n historical samples, we can approximate the optimal spend rates up to an additive factor that goes down at the rate of $\tilde{O}(n^{-1/3})$. In section 5.3.2, we show that in a simpler setting with constant prices, we can obtain a tighter error bound that goes down at the rate of $\tilde{O}(n^{-1/2})$.

5.3.2 Tighter results for Constant Prices

We consider a simpler setting where within an episode the price per impression is fixed as p and only the value is sampled from distribution F_e. For the setting where all prices in episode e are p, the spend function (Definition 5.2.7) simplifies to:

$$\overline{G_e}(\mu) = (1 - F_e((1 + \mu)p)) \cdot p.$$

To estimate $\overline{G_e}(\mu)$ we only need to estimate $F_e((1 + \mu)p)$; we give the procedure ApproxSpendFP in Algorithm 8. The concentration guarantees for $\overline{G_e}(\mu)$ follow from a straightforward application of the DKW inequality (Lemma 5.2.3).

Algorithm 8: ApproxSpendFP: Approximate spend for constant prices.

Input: Number of samples n, $(V_1, \ldots, V_n)$: values samples, p: price of each impression
Goal: Estimate $G(\mu) = \mathbf{E}_{(v)}[\mathbb{1}\{v \geq (1 + \mu)p\}\, p]$
$\widehat{F}(v) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{V_n < v\}$ `// Empirical cdf estimate for values`
$\widehat{G}(\mu) = p(1 - \widehat{F}((1 + \mu)p))$ `// Estimate spend function`
return $\widehat{G}(\mu)$

Lemma 5.3.4. *Given n value samples from F_e and price p, ApproxSpendFP (Algorithm 8) returns the approximate episodic spend function $\widehat{G_e}$ such that with probability at least $1 - \alpha$*

it holds that

$$\left\|\widehat{G_e} - \overline{G_e}\right\| \leq p\sqrt{\frac{\log\frac{2}{\alpha}}{2n}}.$$

Proof. Using the DKW inequality (Lemma 5.2.3), with probability at least $1-\alpha$, we have $\left\|\widehat{F_e} - F_e\right\| \leq \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}$. Consider any $\mu \geq 0$, we have

$$\begin{aligned}
|\widehat{G_e}(\mu) - \overline{G_e}(\mu)| &= |((1-\widehat{F_e}(1+\mu)p) - 1 + F_e((1+\mu)p)) \cdot p| \\
&= p \cdot |\widehat{F_e}((1+\mu)p) - F_e((1+\mu)p)| \\
&\leq p \cdot \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}.
\end{aligned}$$

□

Combining the results of results of Lemma 5.3.4 and Lemma 5.3.2, we can show a tighter analogue of Theorem 5.3.1 for the constant price setting.

Theorem 5.3.5. *Given an episodic setting with fixed prices p and parameters B, T, E, sampling oracles F_e, sampling budget n, K, with probability at least $1-\delta$, for each episode e, by replacing* ApproxSpendSP *(Algorithm 7) with* ApproxSpendFP *(Algorithm 8) (at Line 7),* ApproxSpendRate *(Algorithm 6) returns spend rate $\widehat{\rho}_e$ such that:*

$$|\widehat{\rho}_e - \rho_e| \leq (E+1)p \cdot \sqrt{\frac{\log\frac{2E}{\delta}}{2n}}.$$

5.4 Pacing using Approximate Spend Rates

Now that we have learned the spend rates, in this section we show how we can adapt the Adaptive Pacing Algorithm of [81] to work with changing spend rates $\rho'_1, \cdots, \rho'_E$ which approximate the optimal spend rates.

The main idea is that using our learned spend rates, we can efficiently divide the budget across the episodes and then within each episode, we work with the budget assigned to us, and use the adaptive pacing algorithm of Balseiro and Gur [81] as subroutine. We present

this algorithm as EpisodicAdaptivePacing (Algorithm 9), a detailed version of which appears as Algorithm 16 in Appendix B.2.

Algorithm 9: EpisodicAdaptivePacing

Input: Budget B, rounds T, episodes E, spend plan $(\rho'_1, \ldots, \rho'_E)$, step size η, max shading $\bar{\mu}$.

$\mu_i \leftarrow [0, \bar{\mu}]$, $\textsc{Budget}_1 \leftarrow B$, $\tau \leftarrow \frac{T}{E}$, $\widehat{B}_1 \leftarrow \rho'_1 \cdot \tau$

for $t = 1, \ldots, T$ **do**

  $e \leftarrow \lceil t/E \rceil$

  Observe value v^t

  Post bid $b^t \leftarrow \min\left\{\frac{v^t}{1+\mu^t}, \widehat{B_e}, \textsc{Budget}^t\right\}$

  Observe expenditure z^t

  $\mu^{t+1} \leftarrow \textsc{Proj}_{[0,\bar{\mu}]}[\mu^t - \eta(\rho'_e - z^t)]$

  $\widehat{B}_e \leftarrow \widehat{B}_e - z^t$, $\textsc{Budget}^{t+1} \leftarrow \textsc{Budget}^t - z^t$

  if $t \pmod E = 0$ **then**

    $\widehat{B}_{e+1} \leftarrow \rho'_{e+1} \cdot \tau + \widehat{B}_e$

  end

end

At the beginning of the campaign, we instantiate an overall budget BUDGET as the total budget of the campaign and an episodic budget $\widehat{B}_1$ for the first episode. The budget for each episode is limited ahead of time and if algorithm runs out of the episodic budget $\widehat{B}_e$, then it cannot buy more item in this episode, even though it may have leftover budget for the whole campaign. The intuition behind this is that the budget assigned to each episode is based on the (approximation of) the optimal spend rate. If there is left over budget after an episode ends, then the budget is simply carried forward to the next episode.

In each episode, the adaptive pacing algorithm tries to match the spend in each round to target spend rate of that round. Intuitively the algorithm works by taking the equivalent of a Stochastic Gradient Descent step in the direction of the negative of the gradient of the Lagrangian of that episode. Note that here the Lagrangian dual /average Lagrangian dual for each episode is different as is characterised by the budget for that episode. We can now show that if the spend rate estimates are good, then the resulting strategy has vanishing regret.

Definition 5.4.1 (Admissible Distributions)**.** Joint distribution $\vec{Q}$ such that the dual function

$\Psi_e(\mu, B_e) = \mathbf{E}_{(v,p)\sim Q_e}\left[\tau\left(v - (1+\mu)p\right)^+ + \mu B_e\right]$ is thrice differentiable in μ for all e and B_e with bounded gradients and is strongly convex, where price distribution D_e is atomic with all mass on p, or d_e is Lipschitz and bounded.

Lemma 5.4.1. *If the spend rates used by Algorithm 9 satisfy $\rho_e \geq \rho'_e \geq (1-\omega)\rho_e$, with parameters B, T, E resulting in strategy $\mathcal{A}$, and $\vec{Q}$ satisfies Definition 5.4.1 where $\rho_1, \cdots, \rho_E$ are the optimal spend rates, then setting $\epsilon = \omega$, we have*

$$(1-\epsilon)\text{-}\textsc{Reg}_{\mathcal{A}}(T) \leq \tilde{O}\left(\sqrt{ET}\right).$$

To prove the lemma, we need the following additional result:

Lemma 5.4.2. *If $\rho_e \geq \rho'_e \geq (1-\omega)\rho_e$, it holds that*

$$\inf_{\mu\geq 0} \Psi_e(\mu, \tau\rho'_e) \geq (1-\omega) \inf_{\mu\geq 0} \Psi_e(\mu, \tau\rho_e).$$

where $\Psi_e(\mu, B_e) = \mathbf{E}_{(v,p)\sim Q_e}\left[\tau\left(v - (1+\mu)p\right)^+ + \mu B_e\right]$.

Proof. Let the optimizer of $\Psi_e(\mu, \tau\rho'_e)$ be μ'. We know that the optimizer of $\Psi_e(\mu, \tau\rho_e)$ is $\mu^\star$. Note that since $\rho_e > \rho'_e$, using monotonicity of the spend functions, $\mu^\star < \mu'$ Consider

$$\begin{aligned}
&\Psi_e(\mu', \tau\rho'_e) - (1-\omega)\Psi_e(\mu^\star, \tau\rho_e) \\
&= \tau\mathbf{E}_{(v,p)\sim Q_e}\left[(v-(1+\mu')p)^+ + \mu'\rho'_e - (1-\omega)\left(v-(1+\mu^\star)p\right)^+ - (1-\omega)\mu^\star\rho_e\right] \\
&= \tau\mathbf{E}_{(v,p)\sim Q_e}\left[(\mu'\rho'_e - \mu^\star(1-\omega)\rho_e) + (v-(1+\mu')p)^+ - \left(v-(1+\mu^\star)p\right)^+ + \omega\left(v-(1+\mu^\star)p\right)^+\right] \\
&= \tau\mathbf{E}_{(v,p)\sim Q_e}\left[(\mu'\rho'_e - \mu^\star(1-\omega)\rho_e) - \mathbb{1}\left\{\mu^\star p \leq v - p \leq \mu' p\right\}\left(v-(1+\mu^\star)p\right) + \omega\left(v-(1+\mu^\star)p\right)^+\right] \\
&\geq 0
\end{aligned}$$

□

Proof of Lemma 5.4.1. Let's assume we divide the budget B into budgets B_e for all episodes $e \in [E]$. This results in an online budget constraint bid pacing problem for each individual

episode. Let $\Psi_e(\mu, B_e) = \mathbf{E}_{(v,p)\sim Q_e}\left[\tau\left(v - (1+\mu)p\right)^+ + \mu B_e\right]$ denote the episodic dual function for episode e when the budget for episode e is B_e. Similar to spend functions, if the budget allocation is optimal, that is $B_e = \tau\rho_e$, we can decompose the dual $\Psi(\mu) = \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}[\psi(\mu)]$ across episodes by using episodic dual functions $\Psi_e(\mu, B_e)$.

$$\Psi(\mu) = \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\left(\sum_{t=1}^{T}\left(v^t - (1+\mu)p^t\right)^+\right) + \mu B\right] \tag{5.12}$$

$$= \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}\sim\vec{Q}}\left[\sum_{e=1}^{E}\Big(\sum_{t=(e-1)\tau+1}^{e\tau}\left(v^t - (1+\mu)p^t\right)^+ + \mu\rho_e\Big)\right] \tag{5.13}$$

$$= \tau\sum_{e=1}^{E}\mathbf{E}_{(v,p)\sim Q_e}\left[(v - (1+\mu)p)^+ + \mu\rho_e\right] \tag{5.14}$$

$$= \sum_{e=1}^{E}\mathbf{E}_{(v,p)\sim Q_e}\left[\tau\left(v - (1+\mu)p\right)^+ + \mu\tau\rho_e\right] \tag{5.15}$$

$$= \sum_{e=1}^{E}\Psi_e(\mu, \tau\rho_e) \tag{5.16}$$

Equation 5.13 follows from Equation B.7; and Equation 5.16 follows from the definition of $\Psi_e(\mu, B_e)$. Thus $\Psi_e(\mu, \tau\rho_e)$ is the dual for the episode when the budget B_e for the episode is $\tau\rho_e$.

Let $\Psi_e(\mu_e^\star) = \min_{\mu\geq 0}\Psi_e(\mu_e^\star, \tau\rho_e)$. Using KKT conditions, similar to Equation 5.5, we can show that if $\mu_e^\star > 0$ for all $e \in [E]$, then for all $e \in [E]$

$$\left(\frac{\partial\Psi_e(\mu, \tau\rho_e)}{\partial\mu}\right)_{\mu_e^\star} = \tau\rho_e - \tau\overline{G_e}(\mu_e^\star) = 0$$

$$\implies \rho_e = \overline{G_e}(\mu_e^\star)$$

$$\implies \overline{G_e}(\mu^\star) = \overline{G_e}(\mu_e^\star)$$

Thus $\mu^\star$ satisfies the KKT conditions for $\Psi_e(\mu, \tau\rho_e)$ as well. This furthermore implies that $\mu^\star$ is an optimizer for $\Psi_e(\mu, \tau\rho_e)$. This results in the following conclusion:

$$\Psi(\mu^\star) = \sum_{e=1}^{E}\Psi_e(\mu^\star, \tau\rho_e) = \sum_{e=1}^{E}\Psi_e(\mu_e^\star, \tau\rho_e) \tag{5.17}$$

This implies that if the budget allocation across each episode is $\tau\rho_e$, i.e optimal, then the optimal value of the dual can be obtained by optimizing the dual of each of the episode.

Let the strategy obtained by using our techniques be called $\mathcal{A}$. $\mathcal{A}$ uses spend rates ρ'_e in each episode and assigns budget according to these rates. Once the budget has been divided, the behaviour of $\mathcal{A}$ in each episodes is independent of the other episodes. Hence we can divide the utility obtained by $\mathcal{A}$ across the episodes, i.e. $\pi^{\mathcal{A}}(\boldsymbol{v}, \boldsymbol{p}) = \sum_{e=1}^{E} \pi_e^{\mathcal{A}_e}(\boldsymbol{v}_e, \boldsymbol{p}_e)$.

Where $\pi_e^{\mathcal{A}_e}(\boldsymbol{v}_e, \boldsymbol{p}_e) = \sum_{t=(e-1)\tau+1}^{e\tau} \left[\mathbb{1}\left\{b_{\mathcal{A}}^t > p^t\right\}(v^t - p^t)\right]$ and $\mathcal{A}_e$ is the strategy induced by $\mathcal{A}$ on episode e by limiting the budget for $\mathcal{A}_e$ as $\rho'_e\tau$.

Thus $\mathcal{A}_e$ is just the adaptive pacing strategy given in Balseiro and Gur [81], being run for episode e with spend rate ρ'_e. Since things are i.i.d within the episode, we can directly use the results of [81]. The corresponding dual induced by the episodic sub-problem with budget $\tau\rho'_e$ is

$$\Psi_e(\mu, \tau\rho'_e) = \mathbf{E}_{(v,p)\sim Q_e}\left[\tau\left(v - (1+\mu)p\right)^+ + \mu\tau\rho'_e\right].$$

The expected utility of $\mathcal{A}_e$ in episode e is given by $\mathbf{E}_{(v,p)\sim Q_e}\left[\pi_e^{\mathcal{A}_e}(\boldsymbol{v}_e, \boldsymbol{p}_e)\right]$. We use a corollary of Lemma 5.2.5 which implies that by fixing the budget for episode e as $\tau\rho'_e$, using $\eta = O(\tau^{-1/2})$, we have

$$\inf_{\mu\geq 0} \Psi_e(\mu, \tau\rho'_e) - \pi_e^{\mathcal{A}_e}(\boldsymbol{v}_e, \boldsymbol{p}_e) \leq O(\sqrt{\tau})$$

We know show that if the estimates ρ'_e are good, the optimal of the episodic dual with budget $\tau\rho'_e$ is not too less compared to optimal episodic dual when the budget of the episode is $\tau\rho_e$. Using Lemma 5.4.2, for an episode e

$$(1-\omega)\inf_{\mu\geq 0} \Psi_e(\mu, \tau\rho_e) - \mathbf{E}_{(v,p)\sim Q_e}\left[\pi_e^{\mathcal{A}_e}(\boldsymbol{v}_e, \boldsymbol{p}_e)\right] = O(\sqrt{\tau}).$$

Summing over all rounds and using Equation 5.17 we get,

$$(1-\omega)\inf_{\mu\geq 0}\Psi(\mu)-\mathbf{E}_{(\boldsymbol{v},\boldsymbol{p})}\left[\pi^{\mathcal{A}}\left(\boldsymbol{v},\boldsymbol{p}\right)\right]=\tilde{O}\left(\sqrt{ET}\right).$$

Using weak duality (Equation B.3), we have

$$(1-\omega)\mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\pi^{H}\left(\boldsymbol{v},\boldsymbol{p}\right)\right]-\mathbf{E}_{(\boldsymbol{v},\boldsymbol{p})}\left[\pi^{\mathcal{A}}\left(\boldsymbol{v},\boldsymbol{p}\right)\right]=\tilde{O}\left(\sqrt{ET}\right).$$

□

Putting Everything Together. The final missing component is that the spend rate estimator yields an additive guarantee, while the pacing algorithm expects a multiplicative guarantee. We give a transformation for the the spend plan in Algorithm 10. Lemma 5.4.3 shows that this yields the multiplicative guarantee.

Algorithm 10: End-to-end algorithm

Input: Budget B, rounds T, episodes E, sampling oracles F_e and D_e, per-episode sampling budget n, Kernel K, scalar s, step size η, max shading param $\bar{\mu}$

$(\widehat{\rho}_1,\ldots,\widehat{\rho}_E)\leftarrow$ ApproxSpendRate(B,T,E,F_e,D_e,n,K,s)

$\widehat{\rho}'_e=\frac{(\widehat{\rho}_e+\Delta)B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}$ for all $e\in 1,\ldots E$.

EpisodicAdaptivePacing$(B,T,E,(\widehat{\rho}'_1,\ldots,\widehat{\rho}'_E),\eta,\bar{\mu})$

Lemma 5.4.3. *Given spend rates $\widehat{\rho}_e$ such that $|\rho_e-\widehat{\rho}_e|\leq\Delta$ for all e, then $\widehat{\rho}'_e=\frac{(\widehat{\rho}_e+\Delta)B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}\geq(1-\frac{2\Delta T}{B})\rho_e$ for all e.*

Proof. From the premise it follows that

$$\rho_e\leq\widehat{\rho}_e+\Delta\leq\rho_e+2\Delta. \tag{5.18}$$

Similarly, scaling all sides by the constant $\frac{B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}$, the inequalities continue to hold:

$$\frac{\rho_e B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}\leq\frac{\left(\widehat{\rho}_e+\Delta\right)B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}\leq\frac{\left(\rho_e+2\Delta\right)B}{\sum_e(\widehat{\rho}_e+\Delta)\tau}. \tag{5.19}$$

Using these observations we can derive the multiplicative lower bound:

$$
\begin{aligned}
\widehat{\rho}'_e &= \frac{(\widehat{\rho}_e + \Delta)B}{\sum_e (\widehat{\rho}_e + \Delta)\tau} && \text{(by definition)} \\
&\geq \frac{\rho_e B}{\sum_e (\widehat{\rho}_e + \Delta)\tau} && \text{(by Eq. 5.19)} \\
&\geq \frac{\rho_e B}{\sum_e (\rho_e + 2\Delta)\tau} && \text{(since } \widehat{\rho}_e + \Delta \leq \rho_e + 2\Delta \text{ by Eq. 5.18)} \\
&= \frac{\rho_e B}{\left(\sum_e \rho_e + 2\Delta\right)\tau} \\
&= \frac{\rho_e B}{\left(\frac{B}{\tau} + 2\Delta\right)\tau} && (\tau \textstyle\sum_e \rho_e = B, \text{ Eq. B.7}) \\
&= \frac{\rho_e}{1 + \frac{2\Delta T}{B}} \\
&\geq \left(1 - \frac{2\Delta T}{B}\right)\rho_e.
\end{aligned}
$$

□

We can now restate our main result formally, which follows from Lemma 5.3.4, Lemma 5.3.3, Lemma 5.4.1, and Lemma 5.4.3.

Theorem 5.4.4 (Main Theorem)**.** *Consider the episodic setting with parameters B, T, E, and n samples from $\vec{Q}$ satisfying Definition 5.4.1. Setting $s = O(n^{-1/3})$ and $\eta = O(\tau^{-1/2})$, with probability at least $1 - \delta$, Algorithm 10 has $(1-\epsilon)$-$\textsc{Reg}_{\mathcal{A}}(T) \leq \tilde{O}\left(\sqrt{ET}\right)$ with*

- $\epsilon = \frac{(E+1)pT}{B}\sqrt{\frac{2\log 2E/\delta}{n}} = \tilde{O}(\frac{1}{n^{1/2}})$ *for the constant-price setting, yielding vanishing regret, and*
- $\epsilon = \tilde{O}(\frac{1}{n^{1/3}})$ *and* $\delta = \frac{2E}{n}$ *for the stochastic-price setting, yielding vanishing regret.*

5.5 Slow-moving Distributions

In this section, we consider a setting where the value and price distributions changes at every time step. In this setting, we still consider an advertiser with budget B who participates in T auctions. Each round t has a product distribution $Q^t = F^t \times D^t$, where F^t is the distribution

over impression value $v^t \in [0, h]$ and D^t over the highest competing bid $p^t \in \mathbb{R}^+$. Thus, the T round setting is characterized by distribution $\vec{Q} = (Q_1, \cdots, Q^t)$. We consider settings where this distribution changes slowly over time.

Definition 5.5.1 ($(\zeta, \theta)-$slow-moving distribution)**.** A T round campaign distribution $\vec{Q} = (Q_1 \cdots, Q^t)$ is called $(\zeta, \theta)-$slow moving if for all $t = 1, \cdots, T-1$, we have

$$\left\|F^{t+1} - F^t\right\| \leq \zeta \quad \text{and} \quad \left\|d^{t+1} - d^t\right\| \leq \theta. \tag{5.20}$$

Even though the value and price distributions change in every round, since $\vec{Q}$ is *slow moving*, we can generate approximately accurate spend plans by treating ranges of auctions as episodes. While the distribution in these episodes aren't stationary, the learned spend plan is approximately accurate as the distribution is slow-moving.

Algorithm 11: Spend Prediction and Pacing for Slowly Changing Distribution

Input: Budget B, Total rounds T, Number of episodes to divide into E, Sampling oracles F^t for values and D^t for prices, sampling budget n, Kernel K, scalar s, step size η, max shading param $\bar{\mu}$

Divide T into E episodes of size $\tau = \frac{T}{E}$

Construct episodic sampling oracles $\widetilde{F_e} = \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau} F^t$ and $\widetilde{D_e} = \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau} D^t$

$(\widehat{\rho}_1, \ldots, \widehat{\rho}_E) \leftarrow \mathsf{ApproxSpendRate}(B, T, E, \widetilde{F_e}, \widetilde{D_e}, n, K, s)$

$\widehat{\rho}'_e = \frac{(\widehat{\rho}_e + \Delta)B}{\sum_e (\widehat{\rho}_e + \Delta)\tau}$ for all $e \in 1, \ldots E$.

$\mathsf{EpisodicAdaptivePacing}(B, T, E, (\widehat{\rho}'_1, \ldots, \widehat{\rho}'_E), \eta, \bar{\mu})$

Definition 5.5.2 (Admissible Moving Distributions)**.** Joint distribution $\vec{Q}$ s.t. it satisfies definition 5.4.1 and for any rounds i and j that fall in the same episode, the spend function is strongly monotone, i.e. $(\mu' - \mu)(\overline{G_i}(\mu) - \overline{G_j}(\mu')) > C(\mu' - \mu)^2$ for some constant C.

As per definition 5.2.7, the average spend function for rounds in episode e can be given as $\overline{G_e}(\mu) = \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau} \mathbf{E}_{(v,p)\sim Q^t}[\mathbb{1}\{v \geq (1+\mu)p\}\, p]$ and for the ex-post optimal bidding strategy of bidding $\frac{v^t}{1+\mu^\star}$, we have

$$\tau \sum_{e=1}^{E} \overline{G_e}(\mu^\star) = \tau \sum_{e=1}^{E} \rho_e = B. \tag{5.21}$$

where $\rho_1, \cdots, \rho_E$ are the optimal spend rates. The accuracy of the spend plan now depends on the choice for E and parameters ζ and θ that capture how fast the distribution is changing.

Lemma 5.5.1. *Given B, T, sampling oracles F^t and D^t such that $\vec{Q}$ is $(\zeta, \theta)-$slow moving, number of episodes to break into E, n, K, then with probability at least $1 - \frac{2E}{n}$, using Algorithm 11, in Line 4, ApproxSpendRate (Algorithm 6) returns spend rates $\widehat{\rho}_e$ such that for every episode e,*

$$|\widehat{\rho}_e - \rho_e| \leq (E+1) \cdot \widetilde{O}\left(\frac{1}{n^{1/3}} + \frac{T(\zeta+\theta)}{E}\right)$$

provided d^t is Lipschitz and bounded by setting $s = n^{-1/3}$.

Proof. Consider $\widetilde{F_e} = \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau} F^t$. For any round t belonging to episode e, we have

$$\left\|F^t - \widetilde{F_e}\right\| = \frac{1}{\tau}\left\|\tau.F^t - \sum_{x=\lfloor t/\tau\rfloor*\tau+1}^{(\lfloor t/\tau\rfloor+1)*\tau} F^x\right\| \leq \frac{1}{\tau}\sum_{x=\lfloor t/\tau\rfloor*\tau+1}^{(\lfloor t/\tau\rfloor+1)*\tau} \left\|F^x - F^t\right\| \leq \frac{\zeta\tau}{2}. \tag{5.22}$$

Similarly, for any round t belonging to episode e we have

$$\left\|d^t - \widetilde{d_e}\right\| \leq \frac{\theta\tau}{2}. \tag{5.23}$$

Let $\widehat{F_e}$ be the empirical cdf obtained using n samples from $\widetilde{F_e}$. Using the DKW inequality (Lemma 5.2.3), with probability at least $1-\alpha$, we have $\left\|\widehat{F_e} - \widetilde{F_e}\right\| \leq \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}$. Similarly, let $\widehat{d_e}$ be the kernel density estimate of $\widetilde{d_e}$ obtained using n samples. Using Lemma 5.2.4, with probability at least $1-\frac{1}{n}$, we have $\left\|\widehat{d_e} - \widetilde{d_e}\right\| = \widetilde{O}\left(\frac{1}{n^{1/3}}\right)$. Combining the concentration results with Eq. (5.22) and Eq. (5.23), we that that with probability at least $1-\frac{2E}{n}$, for all episodes e

$$\left\|F^t - \widehat{F_e}\right\| \leq \sqrt{\frac{\log\frac{n}{E}}{2n}} + \frac{\zeta\tau}{2} \quad \text{and} \quad \left\|d^t - \widehat{d_e}\right\| \leq \widetilde{O}\left(\frac{1}{n^{1/3}}\right) + \frac{\theta\tau}{2}. \tag{5.24}$$

Consider the episodic spend function induced by $\widehat{F_e}$ and $\widehat{d_e}$ as $\widehat{G_e}(\mu)$. For all episodes e,

with probability at least $1 - \frac{2E}{n}$, we have

$$
\begin{aligned}
&|\overline{G_e}(\mu) - \widehat{G_e}(\mu)| \\
&= |\frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau}\int_0^h (1 - F^t((1+\mu)p)) \cdot p \cdot d^t(p)\,\mathrm{d}p - \int_0^h (1 - \widehat{F_e}((1+\mu)p)) \cdot p \cdot \widehat{d_e}(p)\,\mathrm{d}p| \\
&\leq \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau} |\int_0^h (1 - F^t((1+\mu)p)) \cdot p \cdot d^t(p)\,\mathrm{d}p - \int_0^h (1 - \widehat{F_e}((1+\mu)p)) \cdot p \cdot \widehat{d_e}(p)\,\mathrm{d}p| \\
&\leq \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau}\int_0^h |(1 - F^t((1+\mu)p)) \cdot d^t(p) - (1 - \widehat{F_e}((1+\mu)p))\widehat{d_e}(p)| \cdot p\,\mathrm{d}p \\
&\leq \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau}\int_0^h \left(|d^t(p) - \widehat{d_e}(p)| + |\widehat{F_e}((1+\mu)p)\widehat{d_e}(p) - F^t((1+\mu)p)d^t(p)|\right) \cdot p\,\mathrm{d}p \\
&\leq \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau}\int_0^h \left(\widetilde{O}\left(\frac{1}{n^{1/3}}\right) + \frac{\theta\tau}{2} + \widetilde{O}\left(\frac{1}{n^{1/3}}\right) + \frac{\theta\tau}{2}\right) \cdot p\,\mathrm{d}p \\
&+ \frac{1}{\tau}\sum_{t=e.\tau+1}^{(e+1)\tau}\int_0^h \left(\sqrt{\frac{\log\frac{n}{E}}{2n}} + \frac{\zeta\tau}{2} + \widetilde{O}\left(\frac{\zeta\tau}{n^{1/3}}\right) + \widetilde{O}\left(\frac{\theta\tau}{n^{1/2}}\right) + \widetilde{O}\left(\frac{1}{n^{5/6}}\right)\right) \cdot p\,\mathrm{d}p \\
&= h \cdot \widetilde{O}\left(\frac{\zeta\tau + 1}{n^{1/3}} + \zeta\tau + \theta\tau\right) \\
&= \widetilde{O}\left(\frac{\zeta\tau + 1}{n^{1/3}} + \zeta\tau + \theta\tau\right)
\end{aligned}
$$

where the first, second, and third inequality follow from triangle inequality. For the fourth step, we use Eq. (5.24) and the fact for any $0 \leq a, b, c, d \leq 1$, $|ab - cd| \leq |c - a| + |d - b| + |(c - a).(d - b)|$. Using Lemma 5.3.2, we get that for all e, with probability at least $1 - \frac{2e}{n}$, we have $|\widehat{\rho}_e - \rho_e| \leq (E + 1) \cdot \widetilde{O}\left(\frac{\zeta\tau}{n^{1/3}} + \frac{\tau(\zeta+\theta)}{E}\right)$.

□

Now that we have approximate spend rates, we can use the estimates to divide the budget across the smaller episodes and use EpisodicAdaptivePacing (Algorithm 9) to perform online pacing. Combining all the guarantees, we can show the following main result for this setting.

Theorem 5.5.2. *For the pacing setting with parameters B, T, $\vec{Q}$, number of episodes to*

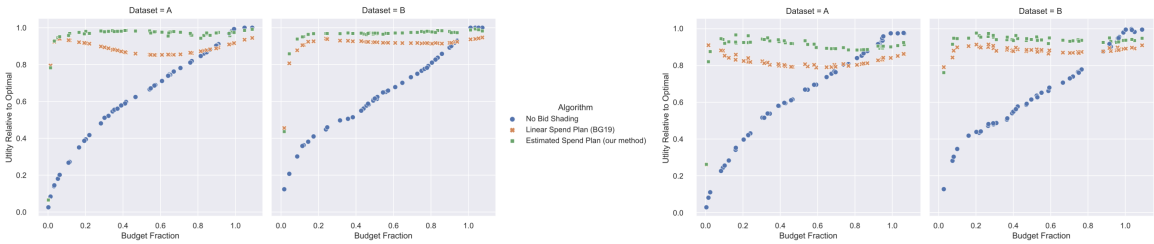
break into E, Kernel K, if $\vec{Q}$ is $(\zeta, \theta)-$slow moving and satisfies Definition 5.5.2, given n samples from $\vec{Q}$, by setting $s = n^{-1/3}$ and $\eta = \tau^{-1/2}$, with probability at least $1 - \frac{2E}{n}$, Algorithm 11 resulting in strategy $\mathcal{A}$ has $(1-\epsilon)$-$\textsc{Reg}_{\mathcal{A}}(T) \leq \tilde{O}\left(\sqrt{ET}\right)$ with $\epsilon = \frac{2ET}{B} \cdot \widetilde{O}\left(\frac{1}{n^{1/3}} + \frac{T(\zeta+\theta)}{E}\right)$.

Theorem 5.5.2 is implied by combining Lemma 5.5.1, Lemma 5.4.3, and Lemma 5.4.1. Theorem 5.5.2 implies our results for the episodic setting can be extended to obtain results for more general settings. We can also observe that in this case, the ϵ in $(1-\epsilon)$-$\textsc{Reg}_{\mathcal{A}}(T)$ doesn't converge to 0 as n grows, since the nonstationarity within an episode does not decrease with more samples.

5.6 Experiments

We now present empirical study of the performance of our proposed algorithm on realistic data from a large online ad platform; further experiments on synthetic data appear in Appendix B.3. We compare the performance of four algorithms, three online and the ex-post optimal benchmark.

- **Estimated Spend Plan (our method).** We implement ApproxSpendSP using the KDE algorithm and the empirical CDF function from statsmodels [117] and run EpisodicAdaptivePacing on the estimated spend plan.
- **Linear Spend Plan.** Given a target budget B, the spend rate in each auction is $\frac{B}{T}$, then run EpisodicAdaptivePacing. The cumulative spend plan is linear and this algorithm is essentially equivalent to the one proposed by Balseiro and Gur [81].
- **No Bid Shading.** The algorithm will bid their value in each auction until they run out of budget.
- **Ex-post Optimal Pacing.** Given realized v^t, p^t for $t \in [T]$, the algorithm buys impressions ordered by bang-per-buck (v^t/p^t) until the budget is spent.



a) Episodic data. a) Non-episodic data.

Figure 5.1: End-to-end performance on realistic datasets.

We compare the utility of the first three algorithms to the ex-post optimal pacing algorithm (the fourth algorithm), reporting the relative utility as a number $\in [0, 1]$.

5.6.1 Datasets

We collected value and price data on a large online advertising platform for 50 campaigns over the course of a single day. We generate instances by taking prices and values and resample and rescale the data at 10 minute intervals to ensure that a fixed bid over that time period yields the same expected value and expected spend on the transformed and original data. To generate an instance, we specify E, T, and n and sample from the transformed dataset.

- **Episodic Data.** From the preprocessed data described above, we generate episodic data (cf. Sections 5.3 and 5.4) as follows: We use $E = 12$ episodes (each representing 2 hours), and let $n = \frac{T}{E} = 10,000$.

- **Non-episodic Data.** For the non-episodic setting (cf Section 5.5) we generate the training and test data differently. For the training data, we use $E_{\text{train}} = 12$ and $n = 12000$. For the online part of the algorithm, we use $E_{\text{online}} = 144$ episodes, $\frac{T}{E_{\text{online}}} = 1000$.

Let $\overline{C}$ be the spend of a campaign which buys all positive utility impressions; for each instance we pick a budget fraction $x \sim U[0, 1.1]$ and set the budget to $x \cdot \overline{C}$.

Table 5.1: Utility relative to the ex-post optimal strategy over all runs of the episodic realistic datasets.

	Our Method	*Linear Spend Plan*
mean	92.5%	89.0%
p10	85.5%	85.1%
p25	94.0%	88.6%
p50	97.4%	92.7%
p75	98.5%	95.2%
p90	99.1%	96.6%

Discussion on the Realistic Data. The realistic instances differ in two ways from the setting for which we have theoretical guarantees. Firstly, there is no guarantee that the data satisfies the "admissible distribution" conditions in Definition 5.4.1. Secondly, like in all realistic data, it may be possible that there are correlations between price and value. For example, if there is a user that is particularly likely to engage with ads, both the advertisers value as well as the competing bids for that user may be high. By studying a setting that captures these real-world properties we aim to shed light on the performance of our approach under realistic conditions.

5.6.2 Results

Episodic Setting In Table 5.1 we give summary statistics over 5 runs for each of the 50 campaigns and in Figure 5.1a we show 50 runs for two chosen representative campaigns. We focus on Table 5.1 first. For each of "Linear Spend Plan" and "Estimated Spend Plan (our method)" we plot summary statistics for the percentage of utility obtained compared to the "Ex-post Optimal Pacing" benchmark. Both Linear Spend Plans and our Estimated Spend Plans capture a large fraction of the optimal utility attainable (on average 89.0% and 92.5%). Our method does beat linear spend plans across the board. In particular the median utility of our method is higher than the 90th percentile for linear spend plans.

For two representative campaigns, we've plotted runs on 50 generated instances in Figure 5.1a. Each plot is a scatter plot for instances generated from a particular campaign.

We've chosen 50 budget fractions uniformly between 0 and 1.1 and generate new training and online data. We run all 4 algorithms on this generated instance, and plot the utility of "no bid shading", "linear spend plan" and "estimated spend plan (our method)" as a fraction of the utility of "ex-post optimal pacing". So each of the 50 instances yields 3 data points on the scatter plot, where the x axis corresponds to the budget fraction and y to the relative utility.

For very small budget fractions, all method tend to do poorly. This is consistent with Theorem 5.4.4 where the error grows as the budget gets smaller. As budgets are bounded away from 0, both linear spend plans and estimated spend plans perform close to optimal, with estimated spend plans outperforming linear spend plans across the board. Recall that for these realistic instances, the distributions are not guaranteed to to satisfy Definition 5.4.1, and that prices and values may be correlated (while learning assumes that prices and values are independent). The fact that our method perform well despite this is promising for deployment in real systems.

Non-episodic Data In the previous section we looked at episodic instances that were generated from realistic data. We now look at what happens when the underlying data is non-episodic, but we estimate a spend plan *as if* the underlying data was episodic. One may do this in the real world where we treat each hour as a separate episode, even when the distribution slowly changes during that hour. For the same campaigns that were shown earlier, we generate data with 144 episodes (so each represents 10 minutes) and estimate a spend plan as if the data came from an instance with 12 episodes (each representing 2 hours). So the estimated spend plan has constant spend for each 2-hour block, even though this consisted of 12 episodes of 10 minutes each. The results are in Figure 5.1b. Both our method and the Linear Spend Plans perform noticeably worse than on episodic data. In particular, the utility of our Estimated Spend Plans is no longer close too 100% of the Ex-post Optimal Pacing utility, but hovers around 90% or so. However, our algorithm still

outperforms the Linear Spend Plan benchmark by quite a margin on a lot of the instances.

Part II

Sequential Decision-Making with Expensive Feedback

CHAPTER 6
ACTIVE ONLINE LEARNING

In this chapter we tackle the challenge of online learning when the true labels are expensive to obtain. We consider the classical problem of multiclass prediction with expert advice, but with an active learning twist. In this new setting the learner will only query the labels of a small number of examples, but still aims to minimize regret to the best expert as usual; the learner is also allowed a very short *burn-in* phase where it can fast-forward and query certain highly-informative examples. We design an algorithm that utilizes Hedge (aka Exponential Weights) as a subroutine, and we show that under a very particular combinatorial constraint on the matrix of expert predictions we can obtain a very strong regret guarantee while querying very few labels. This constraint, which we refer to as ζ*-compactness*, or just compactness, can be viewed as a non-stochastic variant of the disagreement coefficient, another popular parameter used to reason about the sample complexity of active learning in the IID setting. We also give a polynomial time algorithm to calculate the ζ-compactness of a matrix up to an approximation factor of 3. The results presented in this chapter have been published in Kumar *et al.* [20].

6.1 Introduction

The problem of multiclass prediction with expert advice has emerged as a simple yet powerful framework for reasoning about sequential decision tasks. We imagine we have a set of N *experts*, at each round there are K possible outcomes, and where each expert j makes a prediction $X_{t,j} \in [K]$ at time t about an unknown label $y_t \in [K]$. Our learning task is to emit our own estimate $\hat{y}_t \in \Delta_k$ of y_t, that takes into account the advice of each expert along with their historical performance up until this time point. The simple goal is: can we predict well, in the long run, relative to the expert who performs optimally over the

full sequence of predictions, despite that we do not know in advance which expert is best? Moreover, what can we guarantee even when some of these experts may be predicting in an arbitrary or perhaps adversarial fashion? These questions have received a great deal of attention over the past two decades.

The classical algorithm for this problem is commonly known as Hedge [9], although variants are often referred to as *exponential weights* or *weighted majority*. While we give a precise description in Algorithm 12, Hedge is quite simple to explain in words: the algorithm combines the predictions of all the experts on a given round by taking their weighted average, where the weight of an expert exponentially decays according to the number of previous mistakes. Important details must be addressed, such as the exponential decay factor and what to do with fractional predictions, but a great deal of research has made one point very clear: Hedge is essentially the minimax optimal algorithm for the problem of prediction with expert advice.

One of the downsides of Hedge, as with many online learning algorithms, is that it is not *label efficient*: the learning process requires that we observe the target y^t on each round. Obtaining individual labels can, quite often, be very expensive to the learner; indeed this is central to why we design prediction algorithms in the first place. *Active learning*, which refers broadly to a family of frameworks in which the learning algorithm can make selective label queries, are designed precisely with the goal of minimizing the number of needed labels while achieving a suitable learning performance. The key idea is that we do not necessarily need to have a batch of labelled examples prior to training, in many natural scenarios the algorithm may be able to actively engage with the labelling process to query labels on a set of unlabelled examples. The classical Binary Search algorithm is, in some sense, an active learning algorithm to find an element in a sorted list.

It would be hard to argue against the wealth of empirical results showing the benefits of active learning [118, 119, 120, 121, 122]. At the same time, while our theoretical understanding of the label-efficiency gains achieved using this new learning model has been

studied in a range of scenarios [11, 12, 13, 14, 15, 16, 17, 18], our progress towards a full-fledged concrete mathematical foundation of active learning has been relatively slow. A persistent challenge is that precisely identifying scenarios in which active label querying can provide provable benefits, versus those where it necessarily can not, has proven quite difficult [12, 14]. The one notable exception is *disagreement-based* active learning [123]: it has been shown that, as long as the binary hypothesis class possesses a particular property with respect to the underlying probability distribution, known as the *disagreement coefficient*, a recursive algorithm can "zoom in" to the optimal hypothesis and achieve faster learning with lower label complexity. While the disagreement coefficient is somewhat difficult to define, the theoretical work associated to this framework has been perhaps the crowning achievement of the area.

In the following section we give longer outline of the existing work in this area. But it is worth noting up front that nearly all work on active learning has imagined a "batch" setting, where the algorithm is evaluated only at the end of the learning process, in expectation, on new samples. This is surprising, in particular, given that active learning methods are by their nature online, as they seek to iteratively refine their learning process and selection of samples. But thus far there has been no work on putting active learning algorithms to the test in a no-regret setting of prediction with expert advice, where the algorithm's decision is evaluated at each round of the sequence, and where the expert's predictions as well as the labels can be non-stochastic and potentially chosen by an adversary.

In the present paper we aim to remedy this gap, and show that there is a natural framework for active learning in the no-regret setting of prediction with expert advice with strong learning guarantees as well as bounded label complexity. First, we define a notion of complexity of the experts' predictions, somewhat akin to the disagreement coefficient, that provides a key tool in obtaining a provable guarantee; we refer to this as *compactness* for a parameter $\zeta \geq 1$. Quite notably, this quantity can be efficiently estimated up to a constant factor!

Theorem 6.1.1 (Informal)**.** *There is a polynomial time algorithm to calculate the compactness ζ of a matrix up to an approximation factor of 3.*

Second, we define "no-regret active learning" by laying out what we believe is the appropriate analogue to the batch setting. To put it briefly, we imagine a scenario in which the learner must still make sequential predictions on an M-length list of examples, but with the following modifications: (a) the learner is given the sequence of all experts' predictions in advance, (b) the learner can only query the true label y_t on a small number of examples, and (c) the learner is given a very short *burn-in period* where it can "fast-forward" to future rounds in order to query particularly-informative examples. It is this last feature that makes our setting truly *active*, as this term is used in the batch setting, since the learner can recursively seek out useful datapoints. After the short burn-in, however, the learner must play the remainder of the sequence in its original order while querying only a small fraction of the labels.

Third, we propose an online learning algorithm for this setting, ActiveHedge, that leans heavily on Hedge as a subroutine yet uses dramatically fewer label queries. We are able to show the following:

Theorem 6.1.2 (Informal)**.** *Assume we must predict a sequence of labels in $[K]$, we have N experts who have provided predictions (in $[K]$) on all M examples, and the prediction matrix $\mathbf{X} \in [K]^{M \times N}$ is ζ-compact for some $\zeta \geq 1$. If some expert makes only ϵM mistakes, for some $\epsilon > 0$, then with probability $\geq 1 - \rho$ algorithm* ActiveHedge *guarantees that*

1. *with burn-in period of only $O(\zeta \log N \log \frac{1}{\epsilon})$ rounds,*
2. *no more than $O\Big(\zeta \epsilon M \text{polylog}(\frac{N}{\epsilon \zeta \rho})\Big)$ label queries,*
3. *can achieve regret $O\Big(\sqrt{\epsilon M \ln N} + \ln N\Big)$.*

Assuming the prediction matrix $\mathbf{X}$ is ζ-compact for a reasonably-sized constant ζ, this theorem states that the regret of ActiveHedge is indeed *no worse* than Hedge, yet requires a dramatically lower label complexity: roughly $\tilde{O}(\zeta \epsilon M)$ queries are needed. The only extra

power we give the learner is a very brief burn-in period, roughly $\tilde{O}(\zeta)$ rounds, where it can do active exploration of future examples. We now give an illustrative example to view this setting in comparison with more classical batch active learning.

Batch vs Online Active Learning Before we dive into the related work and our results, let us lay out an intriguing scenario. Imagine that a worldwide viral pandemic has recently emerged, and a drug company has been working furiously for months to develop a vaccine to provide immunity to the novel virus. The company has been able to design two candidate vaccines, A and B, has proven to federal regulators that both drugs are safe enough to study in humans, but there's a challenge: some people have a mild allergic reaction to vaccine A but not B, and everyone else has a similar allergic reaction to vaccine B but not A, but this only occurs months after exposure. The company knows that the allergic reaction is based on one of thousands of possible genetic variants, yet must determine quickly which is the relevant gene. Unfortunately there are only two ways to determine if the allergic reaction will occur: (a) wait months to inquire with the patient, or (b) run an expensive test after administering the vaccine that determines immediately whether the allergic reaction will occur.

In this scenario, the "experts" (hypotheses) correspond to candidate genes, a recipient of the vaccine is an example, the true label is their sensitivity to A or B, and the label query cost is incurred by the expensive test needed to detect a future allergic reaction. We introduce this challenge because it helps to highlight the distinction between the two modes of active learning, the classical batch framework and our online setting.

1. If the company decides to take a *batch* active learning approach, they would begin by asking random members of the population to submit their genetic profile and sign up for a vaccine study, but with only a small chance to be selected. The company would then adaptively filter applicants, zero in on particularly-suitable individuals with the relevant genetic information, administer one of the two vaccines, and then immediately

give the expensive test to detect for future allergic reactions. A population-wide vaccine administration protocol can then be developed once the key gene in question is determined.

2. The *online* approach is more aggressive: the company announces that anyone who would like to be vaccinated will have the opportunity, but they must submit a certified genetic profile in advance, arrive at the local mall on a Saturday by 11am, and then wait in a line. All are promised to receive one of the two vaccines, with the goal of minimizing potential allergic reaction; some recipients will be given the expensive test to quickly determine this. Also, all participants are told that *a small number may be brought to the front of the line* so that more medically-informative candidates are treated first; this is the "burn-in" phase which we'll discuss more in Section 6.2.

The typical way that medical procedures are tested and refined is using the first protocol, but we would argue[1] that the second is superior in how it accounts for and manages the costs and benefits of both vaccine recipients and developers. The batch active learning framework has generally been focused on simply minimizing the number of label queries (expensive tests) in order to achieve ϵ accuracy on future examples, but prediction errors that occur in the study phase are not accounted for in the loss objective. The online active learning framework, on the other hand, does not distinguish between study participants and regular vaccine recipients – the goal is simply to induce the least number of allergic reactions at the smallest possible testing cost over the long term.

It is important to note that batch active learning methods, including disagreement-based learning we describe below, can not immediately be applied in the online setting. Batch active learning only considers *label query costs* in the training phase and *prediction error costs* in the testing phase. Another relevant distinction is that our results do not rely on any IID assumption – indeed since the algorithm is allowed to move certain examples ahead in

[1]We want to emphasize that we are **not** proposing to change the drug design and trial framework, as this involves a host of ethical and legal issues not considered here. Rather, drug development provides a useful hypothetical to consider the relative costs of testing and accuracy in an adaptive experimentation problem.

the queue adaptively, new examples are almost certain to be non-independent.

Related Work We briefly survey prior work in the general area of active learning. We will describe salient aspects of these works, and outline how our paper differs from these existing approaches in terms of framework, method, and theory. At a fundamental level, active learning deals with label efficient learning, namely, identifying a good predictor, h_*, from within a hypothesis class, $\mathcal{H}$, based on selectively choosing examples to query for labels. Within this context, a number of methods under a variety of scenarios and assumptions have been studied.

There has been a great deal of work in this area, yet we limit our survey here to a few important themes, in order to draw contrasts and parallels to our setting. Label efficient learning has been considered in pool-based [124, 123], streaming [125, 19, 126] and online scenarios [127, 128, 129]. Pool and stream-based scenarios have been considered largely within the setting of IID examples and/or labels, whereas online methods have been considered under probabilistic [129] as well as adversarial [127] label noise assumptions. A number of approaches including disagreement-based [126, 130, 11, 18, 131, 13], margin-based [132, 133, 134, 135, 136, 12], importance-sampling-based [126, 137], and multiplicative-weight update-based [127] and other online [138, 129] based methods.

In much of the pool and streaming based methods, the underlying assumption is that the examples and labels, are or can be, drawn IID from some fixed unknown distribution, with labels hidden from the learner. The learner after making a number of label requests, not exceeding, say U, outputs a predictor $\hat{h}$. In this line of work, the active-learning protocol is based on comparing $\hat{h}$ against the Bayes optimal predictor on an independent labeled sequence. While there is a rich history of methods, which have been explored under a variety of label noise assumptions, the setting of our work is quite different, in that we make no probabilistic assumptions on the data generation process or label noise; and our active learning protocol, in contrast to these works, does not require independence

between training and test scenarios. In particular, our protocol follows the online regret setting, and the incorrect predictions are penalized on the dataset available to the learner during the training process. On the other hand, our proposed method and theoretical results are fundamentally related to the so called disagreement based methods, and leverages key insights of Hanneke's disagreement coefficient [123]. In particular, we develop the notion of ζ-compactness, which can be interpreted, in some sense, as a deterministic and combinatorial version of disagreement coefficient. Nevertheless, since we make no probabilistic assumptions all previous disagreement-based methods, we cannot leverage classical empirical risk minimization bounds in our context. For this reason, we draw upon insights from the Hedge algorithm and its associated regret bounds, which are agnostic to such probabilistic assumptions.

Our work is also closely related to the label efficient online learning methods, which have been analyzed both under unbiased probabilistic noise as well as adversarial noise assumptions. [1] describes a selective sampling method within the framework of online regret minimization for bounded loss functions. The learner plays M rounds and at time t gets an input x_t, and can decide to seek a label, while being aware of the overall label budget U. Within this setting, leveraging a variant of the Hedge algorithm, and with no additional assumptions on data process, [1] provides regret guarantees, which scale as $M\sqrt{\frac{\log(N)}{U}}$ for N experts (number of hypothesis). A number of online variants to this selective sampling approach have been proposed. [128, 129] introduce probabilistic noise assumptions, and in particular assume that the regression function is linear, and the label noise is unbiased and independent of other examples or queries. The linearity of the regression function together with independent label noise allows them to leverage recursive least-squares techniques. Similar to these works, we also consider a regret-minimization techniques. Different from [128, 129] we make no probabilistic assumptions on label noise. [139, 140] consider the same setting as that of selective sampling where the learner can request the label after making the predictions in each round but don't give any theoretical

guarantees on the label complexity. In contrast to [1] we assume data from all the N rounds are available to the learner a priori. In addition, we impose the notion of ζ-compactness on the dataset of experts' predictions via a concept closely related to disagreement coefficient, which allows for dramatic improvements in label efficiency. As a matter of comparison, say the optimal expert makes ϵM errors, then the existing selective sampling results with budget $U = O(\epsilon M)$, would lead to a regret equal to $\sqrt{\frac{M \log(N)}{\epsilon}}$ in comparison to our result suggesting $\sqrt{\epsilon M \log(N)}$. Nevertheless, improvement in our result can be attributed to the additional imposition of ζ-compactness.

6.2 Notation, Setting, and Background

For the remainder of the paper, we will consider a matrix $\mathbf{X} \in [K]^{M \times N}$ that represent the predictions of a set of N experts on a sequence of M rounds. We will use the notation X_t to refer to the tth row of $\mathbf{X}$, although we will often index rows using the letter i or I. We write $X_{i,j}$ to denote the (i, j)th entry of $\mathbf{X}$. Alongside this matrix will be an (unknown) sequence of labels $y_1, \ldots, y_M \in [K]$. We require a loss function $\ell : \Delta_K \times [K] \to \mathbb{R}$, and for simplicity we restrict our attention to the absolute loss $\ell(\hat{y}, y) := \frac{1}{2}\|\hat{y} - \delta_y\|_1$. Here $\delta_y \in \{0, 1\}^K$ is the indicator vector, with all zeros except a 1 in the y-th coordinate.

6.2.1 Basics: Prediction with Expert Advice, and Hedge

In the classical setting of prediction with expert advice, the learner receives prediction vector X_t at round t, makes a prediction $\hat{y}_t \in \Delta_K$, observes the true label y_t, and suffers the loss $\ell(\hat{y}_t, y_t)$. Each expert j suffers a loss as well, $\ell(X_{t,j}, y_t)$, and note that this loss is conveniently the 0-1 loss as well, $\mathbb{1}_{[X_{t,i} \neq y_t]}$. The algorithm wants to choose the predictions $\hat{y}_1, \ldots, \hat{y}_M$ in order to minimize the *regret*:

$$\textsc{Reg}_{\texttt{alg}} := \sum_{t=1}^{M} \ell(\hat{y}_t, y_t) - \min_{j \in [N]} \sum_{t=1}^{M} \ell(X_{t,j}, y_t).$$

At times it will be convenient to refer to the cumulative loss of expert j as $L_j^M = \sum_{i=1}^M \ell(X_{i,j}, y_i)$. Similarly, the loss of the algorithm is $L_{\text{Hedge}}^M = \sum_{t=1}^M \ell(\hat{y}_t, y_t)$

Algorithm 12: Hedge

Input: $\eta > 0$ /* learning rate parameter */
Init: $\vec{w}^0 = [1, \ldots, 1]$ /* N initial weights */
for $t = 1, \ldots, M$ **do**
  $X_t \leftarrow \texttt{Preds}(t)$ /* Receive expert predictions */
  $\hat{y}_t \leftarrow \text{HedgePredict}(X_t, \vec{w})$
  $y_t \leftarrow \text{QueryLabel}(t)$
  $\vec{w} \leftarrow \text{HedgeUpdate}(\vec{w}, X_t, y_t, \eta)$
end
Procedure HedgePredict$(\vec{x}, \vec{w})$
  $\vec{p} \leftarrow \left[\frac{w_1}{\sum_{i=1}^N w_j}, \ldots, \frac{w_N}{\sum_{i=1}^N w_j}\right]$ /* $\vec{p} \in \Delta_N$ */
  $\hat{y} \leftarrow \vec{p} \cdot \textsc{OneHot}(\vec{x})$ /* Weighted multiclass pred */
  /* OneHot converts multiclass preds $\vec{x} \in [K]^N$ to one-hot matrix encoding $\in (\Delta_K)^N$ */
  return $\hat{y}$ /* $\hat{y}$ is a probability vec in Δ_K */
Procedure HedgeUpdate$(\vec{w}, \vec{x}, y, \eta)$
  /* Decrease weight of incorrect experts */
  for $j = 1, \ldots, N$ **do**
    $w_j^+ \leftarrow w_j \exp(-\eta \mathbb{1}_{[x_j \neq y]})$
  end
  return $\vec{w}^+$

We have already discussed Hedge, the most well-known algorithm for the problem of prediction with expert advice. We lay this out in full detail in Algorithm 12, with two important subroutines, HedgeUpdate and HedgePredict, that will be needed later.

Theorem 6.2.1. *Assume we know a quantity L^* such that $\min_{j=1,\ldots,N} L_j^M \leq L^*$. Then, choosing $\eta = \log\left(1 + \sqrt{\frac{2 \ln N}{L^*}}\right)$ Algorithm 12 guarantees*

$$L_{\text{Hedge}}^M - \min_{j=1,\ldots,N} L_j^M \leq \sqrt{2L^* \ln N} + \ln N. \tag{6.1}$$

This is, in many respects, a fundamental bound. We know, for example, that this can not be made any tighter, even up to constants [141].

6.2.2 Prediction Matrix Compactness

In the typical adversarial learning setting we assume that the experts' predictions and labels are chosen in some arbitrary fashion. On the other hand, it is well understood that to obtain any reasonable learning result in an active label-efficient mode one requires stronger assumptions on the input data. In our framework of prediction with expert advice this will mean we must constrain the matrix $\mathbf{X}$ in an appropriate fashion. Let us now describe a particular condition on $\mathbf{X}$, which we call compactness, that measures a purely combinatorial property of the space of predictions.

Definition 6.2.1. Given $\mathbf{X} \in [K]^{M \times N}$, and for any subset $V \subseteq [N]$ of experts, the *points of contention* of V is the set

$$\textsc{PoC}_{\mathbf{X}}(V) := \{i \in [M] \mid \exists j, j' \in V : X_{i,j} \neq X_{i,j'}\}$$

For any set of experts, the points of contention are the collection of examples where at least two of the experts in the set disagree.

Definition 6.2.2 (ζ- Compactness)**.** For some $\zeta \geq 1$, we say that an expert prediction matrix $\mathbf{X}$ is ζ-*compact* if it satisfies

$$\frac{|\textsc{PoC}_{\mathbf{X}}(V)|}{\max_{j,j' \in V} |\textsc{PoC}_{\mathbf{X}}(\{j, j'\})|} \leq \zeta \tag{6.2}$$

for each $V \subset [N]$ with $|V| \geq 2$. We refer to the *compactness* of $\mathbf{X}$ as the smallest ζ for which inequality (6.2) holds.

Given a prediction matrix $\mathbf{X}$, the compactness of $\mathbf{X}$ controls the divergence between two key quantities of a group of experts V: the total number of points of contention of all of V versus the largest number of points of contention over any pair in the group. In one sentence, the matrix $\mathbf{X}$ is ζ-compact if the size of the contentious set for any subset of experts is never

ζ larger than that of the most contentious pair of experts in it. Here are two illuminating examples that illustrate matrix compactness:

1. Let $K = 2$, $M = N$ and let $\mathbf{X}$ be the identity matrix, with all 0 entries except 1s on the diagonal. The compactness of this matrix is $\frac{M}{2}$, unfortunately, which is very large. That's because if you take $V = [N]$ we see that $\text{PoC}_{\mathbf{X}}(V) = [M]$ the whole set of examples. But for any pair j, j' we have $\text{PoC}_{\mathbf{X}}(\{j, j'\}) = \{j, j'\}$. In other words, any group of experts has as many points of contention as members in the group, but any pair of experts will disagree on only two points. This is indeed a very hard case for active learning, as individual examples are not very informative.
2. Continue to let $M = N$ and now let $\mathbf{X}$ be the upper triangular matrix with all 1s on and above the diagonal, and 0s below. This is a very compact matrix, with $\zeta = 1$! That's because for any subset V we have $\text{PoC}_{\mathbf{X}}(V) = \text{PoC}_{\mathbf{X}}(\{\min(V), \max(V)\})$, i.e. the points of contention in V is identically the points of contention for the largest-index and smallest-index experts in the set.

Following point 1 above, we can give a simple bound on the compactness of any expert prediction matrix $\mathbf{X}$.

Theorem 6.2.2. *For any matrix* $\mathbf{X} \in [K]^{M \times N}$*, for* $M \geq 2$*, the compactness of* $\mathbf{X}$ *is less than or equal to* $\min\{M, N\}$

Proof. If for a set of experts V, if $|V| \leq 2$ then $|\text{PoC}_{\mathbf{X}}(V)| = \text{DIAM}(v)$. Assume V has all unique experts. For any set $V \in [N]$, $|\text{PoC}_{\mathbf{X}}(V)| \leq M$, thus $\zeta \leq M$.

For any V, we show that $|\text{PoC}_{\mathbf{X}}(V)| \leq |V|\text{DIAM}(V)$. Let show this by induction over the size of V. For $|V| \leq 2$, the base cases are direct. Assume that it is true for some V, i.e. $|\text{PoC}_{\mathbf{X}}(V)| \leq |V|\text{DIAM}(V)$. If we add one more expert h to this set, then two cases are possible, a) $\text{DIAM}(V + h) = \text{DIAM}(V)$ or b) $\text{DIAM}(V + h) > \text{DIAM}(V)$.

a) $\text{DIAM}(V + h) = \text{DIAM}(V)$

We can show that $|\text{PoC}_{\mathbf{X}}(V + h)| \leq |\text{PoC}_{\mathbf{X}}(V)| + \text{DIAM}(V)$. If this is not true, i.e. if $|\text{PoC}_{\mathbf{X}}(V + h)| > |\text{PoC}_{\mathbf{X}}(V)| + \text{DIAM}(V)$ then h disagrees with all $j \in V$ on at least

$\textsc{Diam}(V) + 1$ points which are not in $\textsc{PoC}_{\mathbf{X}}(V)$. Thus $\textsc{PoC}_{\mathbf{X}}(h, j) \geq \textsc{Diam}(V) + 1 > \textsc{Diam}(V)$ which would imply $\textsc{Diam}(V + h) > \textsc{Diam}(V)$ which is a contradiction. Thus $|\textsc{PoC}_{\mathbf{X}}(V + h)| \leq |V + h|\textsc{Diam}(V + h)$

b) $\textsc{Diam}(V + h) > \textsc{Diam}(V)$

The extra points added in $\textsc{PoC}_{\mathbf{X}}(V)$ by adding h is bounded by $\textsc{Diam}(V + h)$. We get

$$\begin{aligned}
|\textsc{PoC}_{\mathbf{X}}(V + h)| &\leq |\textsc{PoC}_{\mathbf{X}}(V)| + \textsc{Diam}(V + h) \\
&\leq |V|\textsc{Diam}(V) + \textsc{Diam}(V + h) \\
&\leq |V + h|\textsc{Diam}(V + h)
\end{aligned}$$

This implies for any V, $|\textsc{PoC}_{\mathbf{X}}(V)| \leq \textsc{Diam}(V)|V|$. Since $|V| \leq N$, $\zeta \leq N$. □

Comparison to the Disagreement Coefficient. As we mentioned early in the paper, one of the major theoretical accomplishments in the literature on label-efficient statistical learning is the work on disagreement-based active learning, first introduced by [11] with several followup works [131, 14, 19, 123, 142]. The key quantity of interest in this work is known as the *disagreement coefficient*, a scalar that measures the difficulty of active learning with respect to a particular hypothesis class and data distribution. What was shown all the way back to [11] was that this coefficient controls the label complexity of learning on the given task, and they show several examples where the disagreement coefficient is of reasonable size.

While we developed our notion of compactness independently, and with a different model in mind, we later realized that in the case of binary classification our definition can in some sense be viewed as a "derandomization" of Hanneke's disagreement coefficient; we make this more precise in the proposition below. The compactness ζ of a prediction matrix $\mathbf{X}$ does not depend on any notion of IID sampling from an underlying data distribution, as ζ is purely a combinatorial property of the experts' predictions which could have been adversarially chosen. And, while there is some resemblance between the *burn-in* procedure

in Phase I of ActiveHedge and the A^2 algorithm of [11], our results are not at all comparable: the goal of our work was to produce an algorithm that suffers low regret, as it is forced to make a prediction and suffer loss on each example, and be robust against non-stochastic sequences of data.

Proposition 6.2.2.1. *Consider a binary expert prediction matrix* $\mathbf{X}$ *with compactness* ζ*. Construct a data distribution* D *which generates an* x, y *pair by uniformly sampling* x *as a row of* $\mathbf{X}$ *and let* y *be the corresponding label. We can considers the set of experts as an* N*-sized hypothesis class* $\mathcal{H}$*. Then the disagreement coefficient of* $(D, \mathcal{H})$*, as defined by [11], is* 2ζ *where* ζ *is the compactness of* $\mathbf{X}$*.*

6.2.3 Online active learning with experts

Let us now specify the details of our framework for active learning with expert advice. It can be described in terms of the vanilla Hedge setting, but with three key modifications:

1. The sequence of expert predictions, specified by $\mathbf{X}$, can be precomputed and is given to the learner in advance of the prediction task.
2. The learner aims to make only a small number of label queries, limiting the number of times y_t is observed.
3. We allow a very brief *burn-in* period, which we call Phase I, where the learner can "fast-forward" to act on particular examples, and query their labels, out of turn. In Phase II the learner then plays the remaining points, which are the vast majority, in the order they are given, with the occasional label query if needed.

Modification 1 above is not unusual and arises naturally in settings where the experts are a set of pre-selected deterministic hypotheses, the rounds/examples are given by a queue of contexts/input vectors, and we can pre-evaluate each hypothesis on each context (the vaccine development scenario given in the introduction is another such example). Modification 2 captures the underlying goal that we want to skip the potentially-expensive step of obtaining the correct multiclass label in all but a small fraction of rounds; adding this modification

alone is often referred to as *label efficient online learning*, e.g. [143].

Modification 3 is perhaps the most unusual in the context of adversarial online learning, where one assumes that the learner the sequence of examples and labels is chosen in an adversarial fashion. But we would argue that this is actually necessary to achieve any kind of non-trivial guarantee: without a small number of fast-forward rounds, the adversary can simply postpone all informative examples to the end of the sequence, at which point querying their labels would provide no benefit to the learner. Indeed we show that the burn-in period can be extremely short, no more than roughly $O(\zeta \log N \log \frac{1}{\epsilon})$ where ζ is the compactness of $\mathbf{X}$, in order to obtain *the same regret as* Hedge with vastly fewer label queries (roughly $\tilde{O}(\zeta \epsilon M)$).

Note that if we don't allow a burn in phase, the lower bounds of Cesa-Bianchi *et al.* [1, Theorem 13] apply to the online active learning setting as well. This implies that if we don't allow a burn-in phase, then to guarantee the same $\sqrt{2\epsilon M \ln N}$ regret as Hedge, any algorithm would require at least $\frac{C \cdot M}{\epsilon}$ labels for some constant C. Since $\epsilon \leq 1$, $\frac{C \cdot M}{\epsilon} = \Omega(M)$. Thus, without a *burn-in* period, any algorithm would require $\Omega(M)$ labels to get the same regret guarantee as Hedge. Since Hedge also request $O(M)$ labels, there would be no advantage in using anything other than Hedge.

6.3 Algorithm And Performance Guarantee

Henceforth we will let b denote the index of the best expert, i.e. $b = \text{argmin}_{j \in [N]} L_j^M$, and that the number of mistakes satisfies $L_b^M \leq \epsilon M$.

6.3.1 An Overview of ActiveHedge

We present a multiplicative style algorithm ActiveHedge, described precisely in Algorithm 13. First let us give a high-level intuitive description of the procedure. ActiveHedge is divided into two phases.

1. **Phase I.** This is the so-called burn-in period, where the algorithm can fast-forward

to future examples out of turn. On each such example, the algorithm must still make a prediction, and can then query the label. This phase, while short, is done in small epochs of length $k = O(\zeta \log(N/\rho))$, with a total of $\mathrm{T} = O(\log(1/\epsilon))$ epochs. In a given epoch τ the algorithm has a set of "candidate experts" V^τ who have predicted reasonably well thus far. To reduce the number of candidate experts, the algorithm samples future rounds from the points of contention of V^τ, makes a Hedge prediction on each, and then queries the label. At the end of the epoch the algorithm discards any experts in V^τ whose average error was above a given threshold. On the next epoch we shrink the threshold and consider the new set of candidate experts $V^{\tau+1}$, and sample examples from the new set $\textsc{PoC}_{\mathbf{X}}(V^{\tau+1})$, etc.

2. **Phase II.** At the start of this phase the algorithm has a relatively small set of candidate best experts, V^{T}, that were selected in Phase I, and with high probability b remains in V^{T} and also every expert in V^{T} agrees with b on all but $O(\epsilon M)$ examples. With the burn-in segment over the algorithm now plays the remaining examples, which make up the vast majority, in their original (adversarial) order; rounds played in Phase I are skipped. Uses a very simple prediction strategy:

   (a) if the example i is in $\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, we use Hedge to make a prediction on this example, we query the label y_i, and we do a Hedge update on the weights;

   (b) if $i \notin \textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, we simply use an *arbitrary* expert $j^* \in V^{\mathrm{T}}$ and use X_{i,j^*} as our prediction.

   The choice in condition (b) might seem unusual, but recall that *all experts in* V^{T} *agree* on examples $i \notin \textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$. As long as we did not accidentally evict b from our candidate experts in Phase I, the prediction X_{i,j^*} will match that of $X_{i,b}$. Therefore on these rounds we should suffer no regret.

Algorithm 13: ActiveHedge

Parameters : $\epsilon, \eta, k, \mathrm{T}, \zeta$
Input : $\mathbf{X} \in [K]^{M \times N}$
Initialize : $V^0 \leftarrow [N]$, $t \leftarrow 0$, $\textsc{Done} \leftarrow \emptyset$

/* //// **PHASE I** //// Recursively shrink candidate experts */

for $\tau = 0, \ldots, \mathrm{T}-1$ **do**
  $Z_j^\tau \leftarrow 0 \ (\forall j \in [N])$ /* #errs expert j at epoch τ */
  for $c = 0, \cdots, k-1$ **do**
    $I \sim \textsc{PoC}_{\mathbf{X}}(V^\tau)$ /* Sample w/ replacement */
    if $I \notin \textsc{Done}$ **then**
      $\hat{y}_I \leftarrow \mathsf{HedgePredict}(X_I, \vec{w}^t)$
      $y_I \leftarrow \mathsf{QueryLabel}(I)$
      $\vec{w}^{t+1} \leftarrow \mathsf{HedgeUpdate}(\vec{w}^t, X_I, y_I, \eta)$
      $t \leftarrow t+1$ /* increment hedge update count */
      $\textsc{Done} \leftarrow \textsc{Done} \cup \{I\}$
    end
    $Z_j^\tau \leftarrow Z_j^\tau + \mathbb{1}_{[X_{I,j} \neq y_I]} \ \forall j \in V^\tau$
  end
  $\delta^\tau \leftarrow \dfrac{M}{2|\textsc{PoC}_{\mathbf{X}}(V^\tau)|}\left(\dfrac{1}{2^{\tau+1}\zeta} - \epsilon\right)$ /* Update thresh */
  $V^{\tau+1} \leftarrow \left\{j \in V^\tau : Z_j^\tau / k \leq \delta^\tau\right\}$ /* Shrink V */
end

/* //// **PHASE II** //// Play all remaining rounds */

Select $j^* \in V^{\mathrm{T}}$ arbitrarily
for $i = 1, \ldots, M$ **do**
  if $i \in \textsc{Done}$ **then**
    continue /* skip if example already done */
  else if $i \in \textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$ **then**
    $\hat{y}_i \leftarrow \mathsf{HedgePredict}(X_i, \vec{w}^t)$
    $y_i \leftarrow \mathsf{QueryLabel}(i)$
    $\vec{w}^{t+1} \leftarrow \mathsf{HedgeUpdate}(\vec{w}^t, X_i, y_i, \eta)$
    $t \leftarrow t+1$ /* increment hedge update count */
  else
    $\hat{y}_i \leftarrow \textsc{OneHot}(X_{i,j^*})$ /* use default expert j^* */
    /* One-hot encoding required so that $\hat{y}_i \in \Delta_K$ */
  end
end

6.3.2 Regret and Label Guarantees

We now present the regret and label complexity guarantee for ActiveHedge (Algorithm 13)

Theorem 6.3.1. *Assume we have $\epsilon, \rho > 0$, $\vec{y}$, and ζ-compact matrix $\mathbf{X}$ such that $10\epsilon\zeta \leq 1$ and for some $b \in [N]$ we have $\sum_{i\in[M]} \mathbb{1}_{[X_{i,b}\neq y_i]} \leq \epsilon M$. We set the* ActiveHedge *params*

$$k := \left\lceil 192\zeta \log\left(\frac{N}{\rho}\log\frac{1}{10\epsilon\zeta}\right)\right\rceil, \quad \mathrm{T} := \left\lceil \log\frac{1}{10\epsilon\zeta}\right\rceil \text{ and} \\ \eta := \log\left(1+\sqrt{\frac{2\ln N}{\epsilon M}}\right). \tag{6.3}$$

Then with probability at least $1-\rho$:

1. *the number of calls to* QueryLabel *is no more than*

$$O\left(\zeta\log\left(\frac{N}{\rho}\log\frac{1}{10\epsilon\zeta}\right)\log\frac{1}{10\epsilon\zeta} + \epsilon\zeta M\right)$$

2. *the length of Phase I is no more than $\mathrm{T}k$ which, up to logarithmic terms, is $\tilde{O}(\zeta)$ rounds;*

3. *and finally we have that*

$$\textsc{Reg}_{\textsf{ActiveHedge}} \leq \sqrt{2\epsilon M \ln N} + \ln N.$$

Corollary 6.3.1.1. *If the burn-in phase in* ActiveHedge *is limited to only B rounds, then we can achieve the same regret as Hedge with label complexity $\tilde{O}(B + \frac{M}{2^{B/\zeta}})$.*

Theorem 6.3.1 states that ActiveHedge achieves the same regret guarantee as Hedge with high probability while using considerably less labels. Hedge requires a label complexity of M, where as for a small ϵ and ζ, the label complexity of ActiveHedge is closer to $\tilde{O}(\zeta\epsilon M)$.

Before we give the proof of Theorem 6.3.1, we give a basic sketch of the proof. The basic idea is that we divide the regret analysis and the label complexity analysis into the regret and label complexity of the two phases.

In Phase I, using induction, we show that with high probability, the size of the candidate experts set V^τ shrinks in every round and the best expert is always present in V^τ. After the end of the Phase I, we have narrowed down to the set of candidate experts V^{T} so that with high probability $|\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})| = O(\zeta\epsilon M)$, using compactness, yet still $b \in V^{\mathrm{T}}$. In Phase II we only request the labels for the examples that are in $\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, thus the label complexity of Phase II is bounded by $O(\epsilon\zeta M)$.

Bounding the regret of ActiveHedge is surprisingly easy, since for all examples played in Phase I as well as for those played in Phase II from $\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, we appeal directly to Hedge where we have an optimal bound. In many examples in Phase II, where $i \notin \textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, we make a prediction that (with high probability) agrees with expert b and thus we suffer no regret on these rounds.

It should be noted that even though the guarantees in Theorem 6.3.1 are dependent on the knowledge of ϵ and ζ for initializing the parameters K and T of Algorithm 13, for our proofs to follow through, we just an upper bound on the error rate ϵ of the best expert, and similarly for the compactness ζ. In Theorem 6.4.1, we give a polynomial time algorithm to approximate ζ; this can be used to initialize Algorithm 13. Using $\epsilon' > \epsilon$ in Theorem 6.3.1, we still get the same regret guarantee of $\sqrt{2\epsilon M \ln N} + \ln N$ that still depends on ϵ, but the label complexity will now be $O\left(\zeta \log\left(\frac{N}{\rho}\log\frac{1}{10\epsilon'\zeta}\right)\log\frac{1}{10\epsilon'\zeta} + \epsilon'\zeta M\right)$.

To give the formal proof of Theorem 6.3.1, we need a few preliminary lemmas.

Lemma 6.3.2. *If a set of experts H_1 is a subset of another set of experts H_2, then* $\textsc{PoC}_{\mathbf{X}}(H_1) \subseteq \textsc{PoC}_{\mathbf{X}}(H_2)$

Proof. If $i \in \textsc{PoC}_{\mathbf{X}}(H_1)$, then there exist two experts $j, j' \in H_1$, such that $X_{i,j} \neq X_{i,j'}$. Since $H_1 \subseteq H_2$, $j, j' \in H_2$, hence $i \in \textsc{PoC}_{\mathbf{X}}(H_2)$. □

In each epoch τ of Phase I, we maintain a set of candidate experts V^τ and a set of candidate points $\textsc{PoC}_{\mathbf{X}}(V^\tau)$ we might query the labels for. For ease of notation, let $S^\tau = \textsc{PoC}_{\mathbf{X}}(V^\tau)$, $\textsc{Diam}(V) := \max_{j,j' \in V} |\textsc{PoC}_{\mathbf{X}}(\{j, j'\})|$, and for any experts j, j',

let $\text{dist}(j, j') = |\text{PoC}_{\mathbf{X}}(\{j, j'\})|$.

For the purpose of analysis, we partition the set V^τ into two sets. Let

$$B^\tau = \left\{ j \in V^\tau \mid \text{dist}(b, j) > \tfrac{M}{2^{\tau+1}\zeta} \right\}$$

and also $\overline{B^\tau} = V^\tau \setminus B^\tau$.

Intuitively, B^τ are the experts which are far from the best expert and thus they make more mistakes and we want to remove them. Using an inductive analysis, we will show that in each epoch, with high probability, we can shrink the set of candidate experts, i.e for all τ, $V^{\tau+1} \subseteq \overline{B^\tau}$ and that we never remove the best expert b, i.e $b \in V^{\tau+1}$. For the rest of the section, we set $k = \lceil 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}) \rceil$, $T = \lceil \log \frac{1}{10\epsilon\zeta} \rceil$ and $\eta = \log(1 + \sqrt{\frac{2 \ln N}{\epsilon M}})$

In the following lemma, we show that the size of the set of candidate points sampled from in epoch τ is bounded.

Lemma 6.3.3. *If $V^\tau \subseteq \overline{B^{\tau-1}}$, then $|S^\tau| \leq \frac{M}{2^{\tau-1}}$*

Proof. By definition, $S^\tau = \text{PoC}_{\mathbf{X}}(V^\tau)$. Since $V^\tau \subseteq \overline{B^{\tau-1}}$, using Lemma 6.3.2, $S^\tau \subseteq \text{PoC}_{\mathbf{X}}(\overline{B^{\tau-1}})$. By definition, of $\overline{B^{\tau-1}}$, these experts are at a distance of at most $\frac{M}{2^\tau \zeta}$ from the best expert, the diameter of this set is at most $\frac{M}{2^{\tau-1}\zeta}$. Using definition of $\zeta-$compactness, $|\text{PoC}_{\mathbf{X}}(\overline{B^{\tau-1}})| \leq \zeta \cdot \frac{M}{2^{\tau-1}\zeta} = \frac{M}{2^{\tau-1}}$. Hence $|S^\tau| \leq \frac{M}{2^{\tau-1}}$. □

Now we show that in expectation, any expert in B^τ makes a large number of mistakes in epoch τ which we will use to obtain a high probability bound.

Lemma 6.3.4. *If $b \in V^\tau$ then for any j in B^τ, if Z_j^τ is the number of mistakes made in epoch τ, then $\mathbf{E}\left[Z_j^\tau\right] \geq \frac{k}{|S^\tau|}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M)$*

Proof. Since $j \in V^\tau$ and $b \in V^\tau$, By definition of $S^\tau = \text{PoC}_{\mathbf{X}}(V^\tau)$, if for some i, $X_{i,j} \neq X_{i,b}$, then $i \in S^\tau$. b makes at-most ϵM mistakes, so in the worst case, j can disagree with b on these points and be correct, but it has to be wrong on at least $\frac{M}{2^{\tau+1}\zeta} - \epsilon M$ points in S^τ as it disagrees with b on $\frac{M}{2^{\tau+1}\zeta}$ points in S^τ.

We samples k points from S^τ. Let the examples samples in epoch τ be $(I^1, \cdots, I^k)$, then $Z_j^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}$, $\implies \mathbf{E}\left[Z_j^\tau\right] = \sum_{c=1}^k \mathbf{E}\left[\mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}\right] = \sum_{c=1}^k P[X_{I^c,j} \neq y_{I^c}] \geq \sum_{c=1}^k \frac{1}{S^\tau}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M) = \frac{k}{S^\tau}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M)$ □

Lemma 6.3.5. *If $b \in V^\tau$ and $V^\tau \subseteq \overline{B^{\tau-1}}$ then with probability at least $1 - \frac{\rho|B^\tau|}{N \log \frac{1}{10\epsilon\zeta}}$, $V^{\tau+1} \subseteq \overline{B^\tau}$*

Proof. For a fixed $j \in B^\tau$, by definition the number of mistakes, $Z_j^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c,j} \neq y_{I^c}]}$. The probability that we keep j in $V^{\tau+1}$ is

$$
\begin{aligned}
&P\Big[\frac{Z_j^\tau}{k} \leq \frac{1}{2|S^\tau|}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M)\Big] \\
=&P\Big[\frac{Z_j^\tau}{k} - \frac{1}{|S^\tau|}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M) \leq -\frac{1}{2|S^\tau|}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M)\Big] \\
\leq&P\Big[\frac{Z_j^\tau}{k} - \mathbf{E}\Big[\frac{Z_j^\tau}{k}\Big] \leq -\frac{1}{2|S^\tau|}(\frac{M}{2^{\tau+1}\zeta} - \epsilon M)\Big] \\
\leq&\exp(-\frac{k}{12}(\frac{\frac{M}{2^{\tau+1}\zeta} - \epsilon M}{2|S^\tau|})) \quad \text{(Chernoff Lower tail)} \\
\leq&\exp(-\frac{k}{12}(\frac{1 - 2^{\tau+1}\zeta\epsilon}{8\zeta})) \quad (\text{as } |S^\tau| \leq \frac{M}{2^{\tau-1}}) \\
\leq&\exp(-\frac{k}{12}(\frac{1}{16\zeta})) \quad (\text{as } \tau < \log_2 \frac{1}{10\epsilon\zeta}) \\
=&\frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}))
\end{aligned}
$$

Thus, with probability at least $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$, $Z_j^\tau > \delta^\tau$, thus $j \notin V^{\tau+1}$. A union bound over $j \in B^\tau$ gives the proof. □

So far in the inductive process we have shown that we shrink V^τ to only keep experts from $\overline{B^\tau}$. Now we show that with high probability, we never remove the best expert b.

Lemma 6.3.6. *If Z_b^τ is the number of mistakes made in epoch τ by the best expert b, then $\mathbf{E}[Z_b^\tau] \leq \frac{k\epsilon M}{S^\tau}$*

Proof. Since the best expert makes at-most ϵM mistakes, in the worst case all of these ϵM examples are present in S^τ. Since we samples k points from S^t, $Z_b^\tau = \sum_{c=1}^k \mathbb{1}_{[X_{I^c,b} \neq y_{I^c}]}$ $\implies \mathbf{E}[Z_b^\tau] = \sum_{c=1}^k \mathbf{E}\left[\mathbb{1}_{[X_{I^c,b} \neq y_{I^c}]}\right] = \sum_{c=1}^k P[X_{I^c,b} \neq y_{I^c}] \leq \sum_{c=1}^k \frac{\epsilon M}{S^\tau} = \frac{k\epsilon M}{S^\tau}$ □

Lemma 6.3.7. *If $b \in V^\tau$ and $V^\tau \subseteq \overline{B^{\tau-1}}$ then with probability at least $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$, $b \in V^{\tau+1}$*

Proof. The probability that b is not present in $V^{\tau+1}$ is

$$
\begin{aligned}
& P\Big[\tfrac{Z_b^\tau}{k} \geq \tfrac{1}{2|S^\tau|}(\tfrac{M}{2^{\tau+1}\zeta} - \epsilon M)\Big] \\
=& P\Big[\tfrac{Z_b^\tau}{k} \geq \tfrac{\epsilon M}{2|S^\tau|}(\tfrac{1}{2^{\tau+1}\epsilon\zeta} - 1)\Big] \\
\leq& P\Big[\tfrac{Z_b^\tau}{k} \geq \mathbf{E}\Big[\tfrac{Z_b^\tau}{k}\Big]\tfrac{1}{2}(\tfrac{1}{2^{\tau+1}\epsilon\zeta} - 1)\Big] \\
\leq& \exp(-\frac{k\epsilon M}{6|S^\tau|}\frac{1}{2}(\frac{1}{2^{\tau+1}\epsilon\zeta} - 3)) \quad \text{(Chernoff upper tail)} \\
\leq& \exp(-\frac{k}{3}(\frac{\frac{M}{2^{\tau+1}\zeta} - 3\epsilon M}{2|S^\tau|})) \\
\leq& \exp(-\frac{k}{3}(\frac{1 - 2^{\tau+1}3\zeta\epsilon}{8\zeta})) \quad (\text{as } |S^\tau| \leq \frac{M}{2^{\tau-1}}) \\
\leq& \exp(-\frac{k}{3}(\frac{1}{16\zeta})) \quad (\text{as } \tau < \log_2 \frac{1}{10\epsilon\zeta}) \\
=& \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad (\text{as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}))
\end{aligned}
$$

□

Combining the two results, we can prove the inductive step.

Lemma 6.3.8. *If $b \in V^\tau$ and $V^\tau \subseteq \overline{B^{\tau-1}}$, then with probability at least $1 - \frac{\rho}{\log \frac{1}{10\epsilon\zeta}}$, $b \in V^{\tau+1}$ and $V^{\tau+1} \subseteq \overline{B^\tau}$*

Proof. Union bound over Lemma 6.3.5 and 6.3.7. □

We consider the base case and show that even in the first round, we shrink V^0 to get V^1 and that we don't remove b.

Lemma 6.3.9. *With prob. $\geq 1 - \frac{\rho}{\log \frac{1}{10\epsilon\zeta}}$, $V^1 \subseteq \overline{B^0}$ and $b \in V^1$*

Proof. $\delta^0 = \frac{k}{2}(\frac{1}{2\zeta} - \epsilon)$. For any fixed $j \in B^0$, $\mathbf{E}\big[Z_j^0\big] \geq k(\frac{1}{2\zeta} - \epsilon)$ (6.3.4). Probability that

$j \in V^1$ is

$$
\begin{aligned}
& P\Big[\tfrac{Z_j^0}{k} \leq \tfrac{1}{2}(\tfrac{1}{2\zeta} - \epsilon)\Big] \\
\leq & P\Big[\tfrac{Z_j^0}{k} - \mathbf{E}\Big[\tfrac{Z_j^0}{k}\Big] \leq -\tfrac{1}{2}(\tfrac{1}{2\zeta} - \epsilon)\Big] \\
\leq & \exp(-\frac{k}{12}(\frac{1-2\zeta\epsilon}{4\zeta})) \quad \text{(Chernoff lower tail)} \\
\leq & \exp(-\frac{k}{12}(\frac{1}{8\zeta})) \quad \text{(as } 1-2\zeta\epsilon > 1/2\text{)} \\
\leq & \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad \text{(as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})\text{)}
\end{aligned}
$$

Thus with probability at least $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$, $j \notin V^1$

For b, $\mathbf{E}[Z_b^0] \leq \frac{k}{\epsilon}$. Probability that $b \notin V^1$

$$
\begin{aligned}
& P\Big[\tfrac{Z_b^0}{k} \geq \tfrac{1}{2}(\tfrac{1}{2\zeta} - \epsilon)\Big] \\
\leq & P\Big[\tfrac{Z_b^0}{k} - \mathbf{E}\Big[\tfrac{Z_b^0}{k}\Big] \geq \tfrac{1}{2}(\tfrac{1}{2\zeta} - 3\epsilon)\Big] \\
\leq & \exp(-\frac{k}{3}(\frac{1-6\zeta\epsilon}{4\zeta})) \quad \text{(Chernoff lower tail)} \\
\leq & \exp(-\frac{k}{3}(\frac{1}{8\zeta})) \quad \text{(as } 1-6\zeta\epsilon > 1/2\text{)} \\
\leq & \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}} \quad \text{(as } k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})\text{)}
\end{aligned}
$$

Thus with probability at least $1 - \frac{\rho}{N \log \frac{1}{10\epsilon\zeta}}$, $b \in V^1$

Union bound over $j \in B^0$ and over b proves the statement of the lemma. □

Now that we have proved the inductive step and the base case, we can use these results to state the result for Phase I.

Lemma 6.3.10. *In* ActiveHedge *(algorithm 13), when Phase I ends after* $\mathrm{T} = \frac{1}{10\epsilon\zeta}$ *epochs, with probability at least* $1-\rho$, $b \in V^{\mathrm{T}}$ *and for all* $j \in V^T$, $\mathrm{dist}(b, j) \leq 10\epsilon M$

Proof. Using induction and union bound over $\tau = 1, \cdots, \mathrm{T}$ for Lemmas 6.3.9 and 6.3.8, we get that with probability at least $1-\rho$,

$b \in V^{\mathrm{T}}$, and $V^{\mathrm{T}} \in \overline{B^{\mathrm{T}-1}} \subseteq \left\{ j \in [M] \mid \text{dist}(b, j) \leq \frac{M}{2^{\mathrm{T}}\zeta} \right\}$

Using $\mathrm{T} = \log(\frac{1}{10\epsilon\zeta})$, we get $\frac{M}{2^{\mathrm{T}}\zeta} = \frac{M}{2^{\log(\frac{1}{10\epsilon\zeta})}\zeta} = 10\epsilon M$ □

Now that we have shown that at the end of Phase I, i.e the *burn-in* period, we have considerably shrunk down our set of candidate experts and thus confusing points. We can prove Theorem 6.3.1.

Since, ActiveHedge (Algorithm 13) is divided into two phases, a portion of the regret is incurred in each phase. The examples we predict and request labels for in Phase I are denoted by the set DONE at the end of Phase I. So the portion of regret incurred in Phase I be $\mathrm{R}^{\mathrm{I}} = \sum_{i \in \text{DONE}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$. For Phase II, the points are either in $S^{\mathrm{T}} = \text{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$ where we make hedge updates and request for labels, or they are not in $\text{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, and we use an arbitrary expert $j^* \in V^{\mathrm{T}}$ to make predictions. Let the regret on the points in $\text{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, i.e. the points of contention for V^{T} in phase II be $\mathrm{R}^{\text{con}} = \sum_{i \in ([M] \setminus \text{DONE}) \cap S^{\mathrm{T}}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$ and the total regret for the points in Phase II not in $\text{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$ be $\mathrm{R}^{\text{agree}} = \sum_{i \in ([M] \setminus \text{DONE}) \setminus S^{\mathrm{T}}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i))$

Proof of Theorem 6.3.1. First, let's show the regret bound,

<u>Regret Bound:</u>

Since $\text{REG}_{\text{ActiveHedge}} = \mathrm{R}^{\mathrm{I}} + \mathrm{R}^{\text{con}} + \mathrm{R}^{\text{agree}}$, let's consider the terms individually.

• R^{I} and R^{con}: We are using Hedge (Algorithm 12) to make predictions and make updates. If we re-sample a point for which we have already made a prediction, we do not incur loss on it again. We know that $L_b^M \leq \epsilon M$, hence $L^* = \epsilon M$ is an upper bound on the loss of the best expert in $\mathrm{R}^{\mathrm{I}} + \mathrm{R}^{\text{con}}$ as well. Setting $\eta = \log\left(1 + \sqrt{\frac{2 \ln N}{\epsilon M}}\right)$, we can directly use the

regret bound of Theorem 6.2.1, to show that

$$\begin{aligned}
\mathrm{R}^{\mathrm{I}} + \mathrm{R}^{\mathrm{con}} &= \sum_{i \in \textsc{Done} \cup S^{\mathrm{T}}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i)) \\
&\leq \sum_{i \in \textsc{Done} \cup S^{\mathrm{T}}} \ell(\hat{y}_i, y_i) - \min_{j \in [N]} \sum_{i \in \textsc{Done} \cup S^{\mathrm{T}}} \ell(X_{i,j}, y_i) \\
&\leq \sqrt{2\epsilon M \ln N} + \ln N
\end{aligned}$$

• $\mathrm{R}^{\mathrm{agree}}$: Using Lemma 6.3.10, with probability at least $1 - \rho$, the best expert $b \in V^{\mathrm{T}}$. Since $S^{\mathrm{T}} = \textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, all the experts present in V^{T} agree on $[M] \setminus S^{\mathrm{T}}$. Since $([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}} \subseteq M \setminus S^{\mathrm{T}}$ all the experts in V^{T} agree on all examples in $([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}}$. Thus for all $i \in ([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}}$, for any $j \in V^{\mathrm{T}}$, $X_{i,j} = X_{i,b}$. This is also true for j^* selected before the start of Phase II, We get

$$\begin{aligned}
\mathrm{R}^{\mathrm{agree}} &= \sum_{i \in ([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}}} (\ell(\hat{y}_i, y_i) - \ell(X_{i,b}, y_i)) \\
&= \sum_{i \in ([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}}} \ell(X_{i,j^*}, y_i)) - \ell(X_{i,b}, y_i)) \\
&= \sum_{i \in ([M] \setminus \textsc{Done}) \setminus S^{\mathrm{T}}} \ell(X_{i,b}, y_i)) - \ell(X_{i,b}, y_i)) = 0
\end{aligned}$$

Thus with probability at least $1 - \rho$,

$$\textsc{Reg}_{\mathsf{ActiveHedge}} \leq \sqrt{2\epsilon M \ln N} + \ln N$$

<u>Label complexity:</u>

Let's consider the number of labels requested in each phase.

• Phase I:

Since number of epochs $T = \log \frac{1}{10\epsilon\zeta}$ and in each epoch we request the label for $k = 192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta})$ examples, the number of labels requested is Phase I is at most $192\zeta \log(\frac{N}{\rho} \log \frac{1}{10\epsilon\zeta}) \log \frac{1}{10\epsilon\zeta}$. This is also the size of the *burn-in* period.

• Phase II:

Using Lemma 6.3.10, with probability at least $1-\rho$, for every $j \in V^T$, dist$(b, j) \leq 10\epsilon M$, thus $\textsc{Diam}(V^{\mathrm{T}}) \leq 20\epsilon M$. Using the definition of ζ-compactness,

$$|S^{\mathrm{T}}| = |\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})| \leq \zeta \textsc{Diam}(V^{\mathrm{T}}) \leq 20\epsilon\zeta M$$

. Since we only request labels for the examples in $\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$, the number of labels requested in Phase II is bounded by $|\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})|$, which is less than or equal to $20\epsilon\zeta M$

Hence with probability at least $1-\rho$, the number of labels requested in Phase II is at most $20\epsilon\zeta M$

Combining the label complexity for each of the phase, with probability at least $1-\rho$, the number of labels requested by Algorithm 13 is at most

$$O\left(\zeta \log\left(\frac{N}{\rho}\log\frac{1}{10\epsilon\zeta}\right)\log\frac{1}{10\epsilon\zeta} + \epsilon\zeta M\right)$$

Note that the regret bound and the label complexity result hold simultaneously with probability at least $1-\rho$. □

We can now also prove Corollary 6.3.1.1 using the ideas developed in the analysis of Theorem 6.3.1.

6.3.3 Proof of Corollary 6.3.1.1

Proof. In ActiveHedge (Algorithm 13), in the results of Theorem 6.3.1, the learner is allowed to set the length of the burn-in period itself, i.e. it can decide how many examples that we actually need to actively select and move ahead in the queue. The burn-in phase in Theorem 6.3.1 is set in such a way that it minimizes the overall label complexity of the the algorithm required to get the same regret bound as Hedge.

If instead of giving the learner the freedom to set its own length of Phase I, if the learner is only given a budget B of number of examples it can move ahead in the queue, then by

setting $k = \tilde{O}(\zeta)$ and $\mathrm{T} = B/k$, the size of the burn-in phase becomes B. At the end of Phase I, in this case, the size of the set of points of contentions, that is $|\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})|$ is $\tilde{O}(\frac{M}{2^{B/\zeta}})$ (Lemma 6.3.3). Thus, the total samples queried would be $\tilde{O}(B + \frac{M}{2^{B/\zeta}})$.

Similar to Theorem 6.3.1, since we don't make any mistakes on the points outside $\textsc{PoC}_{\mathbf{X}}(V^{\mathrm{T}})$ in Phase II, the number of mistakes is bounded by the mistakes made by Hedge, resulting in the same regret guarantee.

□

6.4 Calculating compactness

Algorithm 14: Calculate compactness

```
Input: X ∈ [K]^{M×N}                                   /* Expert prediction matrix */
Init: ζ̃ ← 0
for all pairs j, j' ∈ [N] do
    V_{j,j'} ← {j, j'}                                 /* Initialize V_{j,j'} */
    /* Add experts with distance from j ≤ dist(j, j') */
    V_{j,j'} ← V_{j,j'} ∪ {h | dist(h, j) ≤ dist(j, j')}
    /* Add experts with distance from j' ≤ dist(j, j') */
    V_{j,j'} ← V_{j,j'} ∪ {h | dist(h, j') ≤ dist(j, j')}
    ζ_{j,j'} ← |PoC_X(V_{j,j'})| / DIAM(V_{j,j'})
    /* Update ζ̃ if a bigger ratio is found */
    if ζ_{j,j'} > ζ̃ then
        ζ̃ ← ζ_{j,j'}
    end
end
Return: ζ̃
```

The compactness of an expert prediction matrix is a combinatorial quantity which is easy to compute for some concept classes, but in the worst case it might be hard to compute exactly as we have a supremum over all subsets of experts. We present an algorithm that gives a 3-approximation of the compactness in polynomial time.

For the remainder of this section and the appendix, for any $V \subset [N]$ let $\textsc{Diam}(V) := \max_{j,j' \in V} |\textsc{PoC}_{\mathbf{X}}(\{j, j'\})|$ and for any experts j, j', let $\mathrm{dist}(j, j') = |\textsc{PoC}_{\mathbf{X}}(\{j, j'\})|$.

Theorem 6.4.1. *If the input matrix* $\mathbf{X}$ *to Algorithm 14 is* ζ*-compact, then Algorithm 14 returns* $\tilde{\zeta}$ *such that* $\frac{\zeta}{3} \leq \tilde{\zeta} \leq \zeta$ *in runtime* $O\left(N^4 M\right)$

Proof. Consider the subset

$$V^* = \underset{V, \textsc{Diam}(V)>0}{\text{argmax}} \frac{|\textsc{PoC}_{\mathbf{X}}(V)|}{\textsc{Diam}(V)}$$

Let $h_1, h_2 \in V^*$ be the experts such that dist$(h_1, h_2) = \textsc{Diam}(V^*)$. For any $h' \in V^*$, dist$(h', h_1) \leq \textsc{Diam}(V)$ and dist$(h', h_2) \leq \textsc{Diam}(V)$, hence $h' \in V_{h_1,h_2}$, i.e $V^* \subseteq V_{h_1,h_2}$ in Algorithm 14. This gives us that $|\textsc{PoC}_{\mathbf{X}}(V_{h_1,h_2})| \geq |\textsc{PoC}_{\mathbf{X}}(V^*)|$

Since we include all experts that are at a distance of at most dist(h_1, h_2) from h_1 or h_2, the diameter $\textsc{Diam}(V_{h_1,h_2}) \leq 3\text{dist}(h_1, h_2) = 3\textsc{Diam}(V^*)$

Using these two facts, we get $\frac{|\textsc{PoC}_{\mathbf{X}}(V_{h_1,h_2})|}{\textsc{Diam}(V_{h_1,h_2})} \geq \frac{|\textsc{PoC}_{\mathbf{X}}(V^*)|}{3\textsc{Diam}(V^*)} = \frac{\zeta}{3}$

We consider all pairs of experts in Algorithm 14, hence the $\tilde{\zeta}$ returned satisfies

$$\tilde{\zeta} \geq \frac{|\textsc{PoC}_{\mathbf{X}}(V_{h_1,h_2})|}{\textsc{Diam}(V_{h_1,h_2})} \geq \frac{\zeta}{3}$$

For the upper bound, since the $\tilde{\zeta}$ returned is $\frac{|\textsc{PoC}_{\mathbf{X}}(V_{j,j'})|}{\textsc{Diam}(V_{j,j'})}$ for some j, j', it is obvious that

$$\tilde{\zeta} \leq \max_{V, \textsc{Diam}(V)>0} \frac{|\textsc{PoC}_{\mathbf{X}}(V)|}{\textsc{Diam}(V)} = \zeta$$

The run time comes from the fact that we consider all $O(N^2)$ pairs of experts and for any subset $V \subseteq [N]$, $|\textsc{PoC}_{\mathbf{X}}(V)|$ can be computed in $O(|V|M)$ and $\textsc{Diam}(V)$ can be computed in $O(|V|^2 M)$ □

As stated earlier, for initializing Algorithm 13 for the results in Theorem 6.3.1, we just need an upper bound on the ζ-compactness. Using Algorithm 14, we can obtain an estimate $\hat{\zeta} = 3\tilde{\zeta}$ such that $\zeta \leq \hat{\zeta} \leq 3\zeta$.

6.5 Experiments

We provide preliminary experiments to compare ActiveHedge (Algorithm 13), with standard Hedge (Algorithm 12) and the label efficient algorithm given by Cesa-Bianchi *et al.* [1]. We first present results on synthetic data in Section 6.5.1 and then provide more evidence on realistic datasets MNIST [144] and CIFAR-10 [145] in Section 6.5.2

6.5.1 Results on synthetic data

For synthesis data, we create synthetic prediction matrices by considering different hypothesis classes for experts.

We consider three different classes of experts for our experiments. In Fig. 6.1: a) we consider linear classifiers passing through the origin as experts. We uniformly N sample linear classifiers from a unit sphere centred at origin. We then sample M points from a unit sphere and classify each point using the N experts to create the expert prediction matrix $\mathbf{X}$. Similarly, in Fig. 6.1: b), we consider multi-dimensional thresholds as experts where a point $x \in \mathbb{R}^d$ is labeled 1 by an expert $h \in \mathbb{R}^d$ if $x_i \geq h_i \forall i \in [d]$. The experts are sampled by sampling thresholds uniformly between 0 and 1. In both the cases, ActiveHedge is able to achieve similar accuracy to Hedge and achieves better performance than Cesa-Bianchi *et al.* [1] in terms of both regret and label complexity.

We also consider the more adversarial case in Fig. 6.1: c), where the expert prediction matrix has an identity matrix like structure with $\zeta = O(N)$. Here the expert prediction matrix is designed such that only one of the experts labels each point as 1, and every expert classifies approximately equal number of points as 1. Even in this adversarial case where the ζ compactness is very high, ActiveHedge out performs the competition. Thus happens because even though the ζ compactness is high, it also implies that by removing an expert from consideration, we also remove a significant fraction of points we are confused on. This allows us to quickly converge to the optimal expert. All experiments are repeated 100 times,

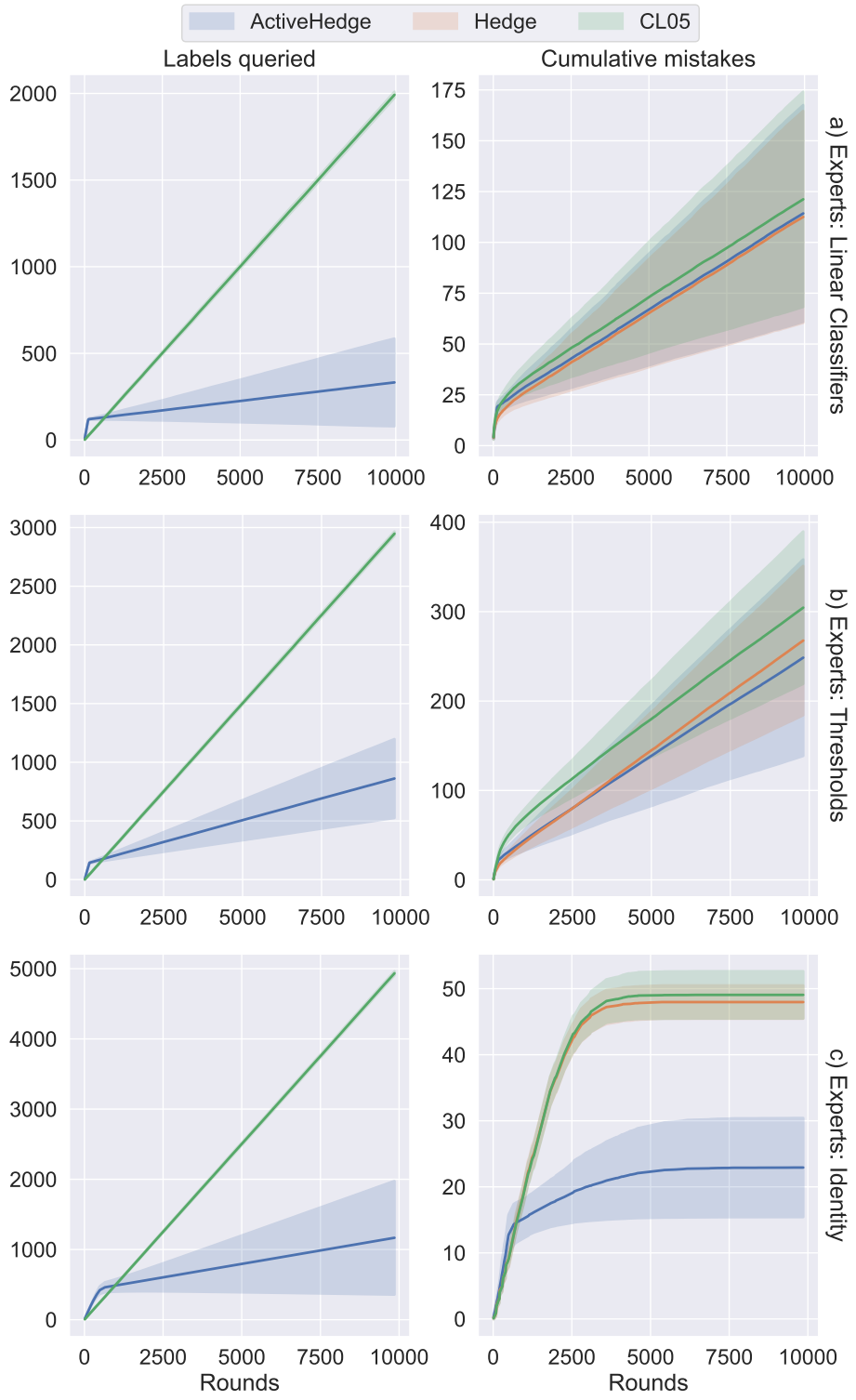


Figure 6.1: Labels queried and the cumulative mistakes of ActiveHedge, Hedge, and Cesa-Bianchi *et al.* [1](CL05) on 3 different synthetic datasets. Hedge queries label in every round and is not shown in Labels queried plots to maintain readability.

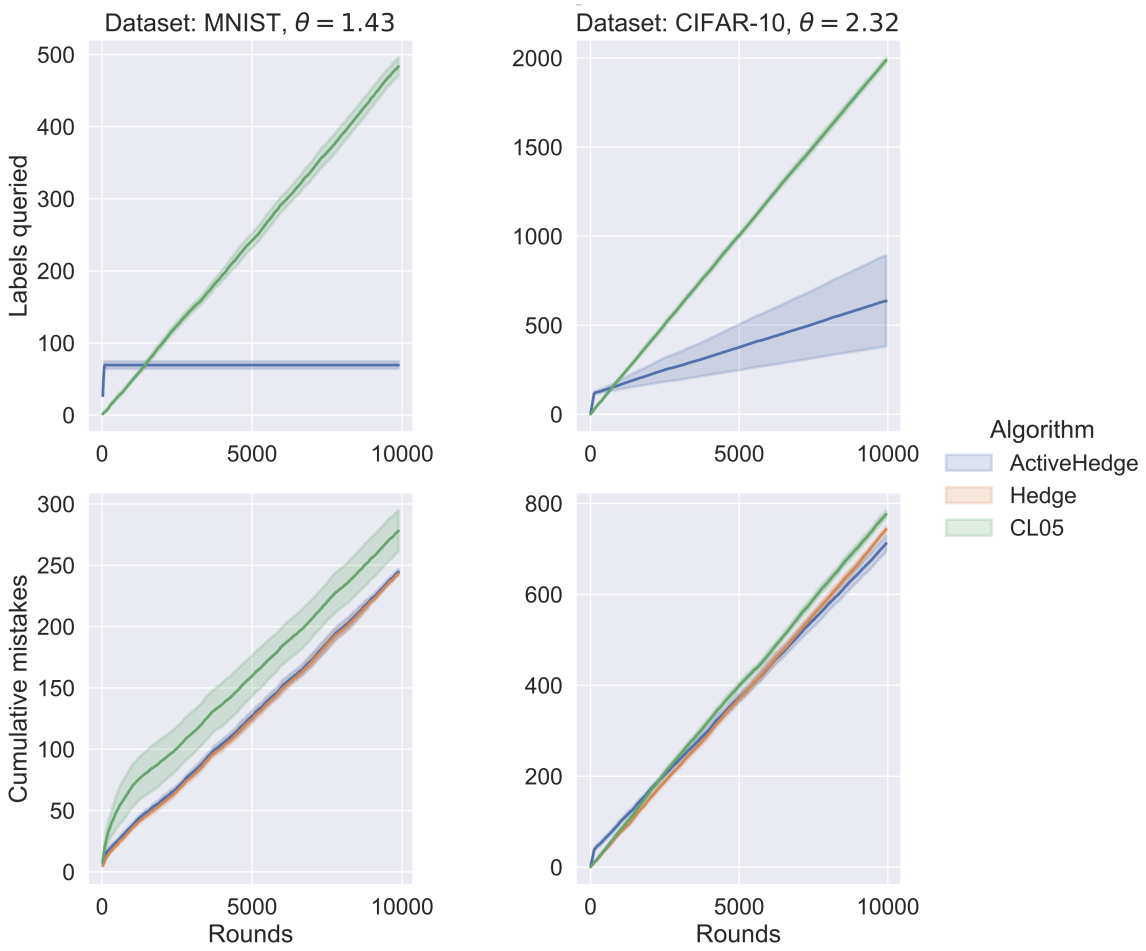


Figure 6.2: Labels queried and the cumulative mistakes of ActiveHedge, Hedge, and Cesa-Bianchi *et al.* [1](CL05) in on real datasets. The sub-figures on the left are the results on MNIST's test-dataset where each expert is small random forests made of small depth trees trained on MNIST's train-dataset. Similarly, the results in right sub-figure are on CIFAR-10's test-dataset where each expert is a convolutions neural network trained on CIFAR-10's train-dataset. Hedge queries label in every round and is not shown in Labels queried plots to maintain readability.

with $M = 10000$ and $N = 100$ and $d = 10$. We use upper bounds for ζ and ϵ and other parameters are set optimally. For all these experiments, ActiveHedge required less than 10% of the labels with the burn-in phase being less than 2% of the points.

6.5.2 Results on realistic data

For experiments on realistic dataset, similar to the results on synthetic data, we use different classes of experts to create our prediction matrices but instead of artificially generating points, we consider real datasets (MNIST [144] and CIFAR-10 [145]) and experts as models trained specifically on these datasets. Both MNIST and CIFAR-10 have 50000 images in the train split, and 10000 images in the test split.

For MNIST, we train 10 different random forests with small depth trees (2 to 12 trees, with depth 3) on the train set and use these trained classifiers as experts to create the expert prediction matrix on the test split of MNIST. Thus, $M = 10000$ and $N = 10$ for this experiment.

Similary for CIFAR-10, we use 10 pre-trained Convolutional neural network models of different depths trained on CIFAR-10 and use the test split for CIFAR-10 to create the expert prediction matrix with $M = 10000$ and $N = 10$ for this experiment as well. We repeated both the experiments 10 times and report the results in Fig. 6.2.

The left sub-figures of Fig. 6.2 represent the result on MNIST dataset and we observe that ActiveHedge (13) converges on the best expert in the *burn-in* phase itself and thus never queries any more labels after the burn-in phase. The regret of ActiveHedge is also similar to the regret of Hedge which queries the labels in every round. We also calculate the exact ζ-compactness of the prediction matrix and in this case, the matrix is only 1.43 (shows as θ in Fig. 6.2).

Similary, the right sub-figures of Fig. 6.2 represent the result on CIFAR-10 which is a more complicated datasets than MNIST. In this experiment as well the regret of ActiveHedge is also similar to the regret of Hedge and the label efficient learner given in Cesa-Bianchi

et al. [1], but the label compolexity is much smaller than either of the competition. For this experiment as well, the prediction matrix had a very small ζ-compactness of 2.32. The small values of ζ-compactness in both the experiments also validates our assumption that ζ can be thought of as a small constant in many realistic settings.

Part III

Sequential Decision-Making in a Limited Feedback Environment while Ensuring Fairness

CHAPTER 7

GROUP FAIRNESS OF EXPOSURE IN BANDITS

In this chapter, we consider the fairness aspect of sequential decision-making. Specifically, we consider a new group-based notion of fairness that promotes exposure in multi-arm bandits. Instead of the usual goal that maximizes reward, this new notion of fairness in exposure ensures that all groups or individuals get opportunities based on their actual quality. This chapter is based on an ongoing project [22].

7.1 Introduction

In the traditional multi-arm bandit setting that we introduced in Chapter 3, there are K arms $a \in [K]$ with mean rewards $\mu_a^* \in [0, 1]$. In each round $t = 1, \cdots, T$,

- Learners sets a distribution $\pi^t \in \Delta_K$ over the K arms
- Learner selects arm $a^t \sim \pi^t$
- Learner receives reward r^t with $E[r^t] = \mu_{a^t}^*$

The standard goal for a learner in a multi-arm bandit setting is to maximize the total expected reward, i.e. $\sum_{t=1}^T \mu_{a^t}^*$ which is equivalent to minimizing *regret*, i.e. $\max_{a \in [K]} T\mu_a^* - \sum_{t=1}^T \mu_{a^t}^*$.

Recent work of Wang *et al.* [21] introduced a new goal that promotes more fair choices by the learner that just aiming to maximize reward. Wang *et al.* [21] introduce merit based fairness-of-exposure in the stochastic multi-armed bandit setting, but at an individual fairness level. In this setting, the learner wants to learn a strategy π^* (which is a distribution over

arms) such that for any two arms a and a' with mean reward μ_a^* and $\mu_{a'}^*$, we have

$$\frac{\pi^*(a)}{f(\mu_a^*)} = \frac{\pi^*(a')}{f(\mu_{a'}^*)}. \tag{7.1}$$

Here, $f : \mathbb{R} \to \mathbb{R}_{>0}$ is a monotonically increasing function which the learner knows and an example could be $f(\mu) = \mu$, in which case we want arms to be selected with probability proportional to their mean reward. Given the true means, the strategy π^* is uniquely determined by Eq. (7.1) [21, Theorem 3.1.1].

Consider a scenario where a company A and company B have two products, let's say cellphones listed on a marketplace. Every time a user searches for "cellphone" on the platform, the platform selects one of the cellphones listed on the store as the top recommendation. Here we can assume that given a product is selected as a response to the search query, there is a stochastic reward that the platform receives by selecting that product. This stochastic reward encapsulates whether the chosen product was sold or not, or if, how much revenue or profit the platform made. Thus, this setting can be modeled as a stochastic multi-arm bandit problem where each product has an expected reward. Let's consider a case that one of the products of A has an expected reward 0.51 and the product of B has an expect reward of 0.49. If we follow the standard notion of reward maximization, the ideal distribution would be to pick the product of company A in each round. Even though the products of companies A and B are very similar in terms of expected reward, but the product of company A gets all of the exposure. This might not be considered a very fair outcome, and the problem is exacerbated when we are talking about people and not products.

Under the fairness notion given by Wang *et al.* [21], assuming $f(x) = x$ for now, in the case we just described, the ideal distribution π^* would ensure that product of company A gets 51% of the time and product from company B gets picked 49% of the time, which could be considered fairer.

Now consider the situation where B knows that the marketplace is using a strategy that

gives fairness in exposure to products based on their quality. B tries to game the system and launches a new product, which is very similar to their old product but is considered an arm for the learner. Now A has one product with a 0.51 expected reward and B has two products listed with 0.49 expected reward each. Under the fairness definition given by Wang *et al.* [21], B would now get about 66% of the total exposure whereas the A would only get 33%. Thus this notion of fairness is prone to manipulation.

To overcome this issue, want to consider a group-based extension of the fairness goals where each arm belongs to a group.

7.2 Setting

We give a description of the ideal group constraints that a policy π must satisfy to be considered *fair*. We call this setting the group fairness of exposure in multi arm bandits setting.

Formally, generalizing the model of [21], there are K arms $a \in [K]$. The average mean reward of an arm a be $\mu_a^* \in [0, 1]$. There are m groups and each arm a belongs to exactly one group $G \subseteq [K]$, given by $G(a)$. Let $\mathcal{G}$ be the set of groups. Let $\vec{\mu_G^*} = (\mu_a^*)_{a \in G}$ be the vector of all the mean rewards for arms inside the group. At the group level we have a group aggregate function $g : [0, 1]^* \to \mathbb{R}$. For the optimal stationary policy π^*, we want that the probability that the selected arm belongs to this group G is proportional to $g(\vec{\mu_G^*})$, that is for any two groups G and G', we want that

$$\frac{P_{a \sim \pi^*}[a \in G]}{g(\vec{\mu_G^*})} = \frac{P_{a \sim \pi^*}[a \in G']}{g(\vec{\mu_{G'}^*})}.$$

We also have arm specific functions f and for a and a' belonging to the same group (that is $G(a) = G(a')$) with individual means μ_a^* and $\mu_{a'}^*$, we want

$$\frac{\pi^*(a)}{f(\mu_a^*)} = \frac{\pi^*(a')}{f(\mu_{a'}^*)}$$

Similar to the arm based fairness setting, given the true means, the optimal policy is one fixed policy that can be calculated as follows:

Fact 7.2.1. *Given mean rewards μ_a^*, functions $g(\cdot)$, $f(\cdot)$, and group identities $G(\cdot)$, the fair policy π^* is given by*

$$\pi^*(a) = \frac{g(\vec{\mu^*_{G(a)}})}{\sum_{G' \in \mathcal{G}} g(\vec{\mu^*_{G'}})} \cdot \frac{f(\mu_a^*)}{\sum_{a' \in G(a)} f(\mu_{a'}^*)}$$

We can also show that the group fairness of exposure setting

Lemma 7.2.2. *The group fairness of exposure in multi arm bandits the setting proposed in Wang* et al. *[21], as when $g(\vec{\mu^*_G}) = \sum_{a \in G} f(\mu_a^*)$, we recover their setting.*

To measure how fair a policy is, we define a notion of fairness regret measures the errors made by the learner in deciding it's policy π^t. Since a policy is uniquely defined by the means of the arm, we define the regret as a function of the policy made using mean vector μ^t as

7.2.1 New Fairness Regret Definition

To weigh both group and individual fairness constraints, an intuitive *fairness regret* definition of algorithm $\mathcal{A}$ is

Definition 7.2.1 (Fairness Regret)**.** For an algorithm $\mathcal{A}$ that chooses the fair policy given by fixing the mean reward estimate as μ^t in round t, the fairness regret over T rounds is defined as:

$$\text{FR}_{\mathcal{A}}^T = \sum_t^T \left(\sum_{G \in \mathcal{G}} \left(|\Pi(G, \mu^t) - \Pi(G, \mu^*)| + \Pi(G, \mu^*) \cdot \sum_{a \in G} |\Gamma(a, \mu^t) - \Gamma(a, \mu^*)| \right) \right)$$

where

$$\Pi(G,\mu) := P_{a\sim\pi_\mu}[a \in G] = \frac{g(\vec{\mu_G})}{\sum_{G'\in\mathcal{G}} g(\vec{\mu_{G'}})}$$

and

$$\Gamma(a,\mu) := P_{x\sim\pi_\mu}[x = a | x \in G(a)] = \frac{f(\mu_a)}{\sum_{a'\in G(a)} f(\mu_{a'})}$$

The $\sum_{G\in\mathcal{G}} (|\Pi(G,\mu^t) - \Pi(G,\mu^*)|)$ part represent the group fairness part of the regret. $\sum_{a\in G} |\Gamma(a,\mu^t) - \Gamma(a,\mu^*)|$ is the individual fairness part of the regret for a group G which is weighed by $\Pi(G,\mu^*)$ which is probability of that group being selected by the optimal policy π^*.

The learn also wants to maximize the reward, but only with respect to the optimal fair policy π^*. Thus, we have a new definition for reward regret which measures the difference between an algorithm's expect reward and the reward of the optimal fair policy π^*

Definition 7.2.2 (Reward Regret)**.** For an algorithm $\mathcal{A}$ that chooses the fair policy given by fixing the mean reward estimate as μ^t in round t, the reward regret over T rounds is defined as:

$$\text{RR}_{\mathcal{A}}^T = \sum_{t=1}^{T} \sum_{a\in[K]} \pi^*(a)\mu_a^* - \sum_{t=1}^{T} \sum_{a\in[K]} \pi^t(a)\mu_a^*$$

Lower bound Since our setting is a generalization of [21]'s setting, the $\Omega(\sqrt{T})$ lower bound still applies for both reward and fairness regret. It is not clear if this lower bound is tight.

Lemma 7.2.3. *Lower bound Since our setting is a generalization of [21]'s setting, the $\Omega(\sqrt{T})$ lower bound still applies for both reward and fairness regret.*

Assumption 7.2.4. *g is a scalar function of average means in the group*

let us assume g is a scalar function of the average mean reward of the arms inside the group, i.e. $\mu_G^* = \frac{\sum_{a\in G}\mu_a^*}{|G|}$. Similar to assumptions of Wang *et al.* [21], for the rest of the chapter, we assume that f is L-Lipschitz and g is L_g-Lipschitz and lower bounded by γ_g.

lower bounded by γ and g is a scalar function of the average mean rewards, i.e , is L_g-Lipschitz and lower bounded by γ_g

7.2.2 Explore then Exploit Algorithm

We first present a simple explore then exploit style algorithm that explores each arm for K rounds then at the end of the explore phase, it uses the empirical means to construct the optimal policy using Fact 7.2.1.

Algorithm 15: Explore then Exploit

Input: Number of arms K, rounds T, functions $f(\cdot), g(\cdot)$, group identities $G(\cdot)$, N

for $t = 1, \cdots, KN$ **do**

  if $t \leq KN$ **then**

    Select arm $a^t = t \bmod K$

  else

    $\forall a, \mu_a^t = \hat{\mu}_a = \sum_{\tau=1}^{KN} \mathbf{1}_{\{a^\tau = a\}} r^\tau / N$

    $\pi^t(a) = \frac{g(\vec{\mu^t_{G(a)}})}{\sum_{G' \in \mathcal{G}} g(\vec{\mu^t_{G'}})} \cdot \frac{f(\mu_a^t)}{\sum_{a' \in G(a)} f(\mu_{a'}^t)}$

    Select arm $a^t \sim \pi^t$

  end

  Receive reward r^t

end

Theorem 7.2.5. *If f is L-Lipschitz, lower bounded by γ and g is a scalar function of the average mean rewards, is L_g-Lipschitz and lower bounded by γ_g, using an explore-then-exploit algorithm (Algorithm 15) with a fixed exploration of $N = \tilde{O}(T^{2/3})$, the fairness regret satisfies the following upper bound:*

$$E[FR_{\mathcal{A}}^T] \leq 2T^{2/3}(K \log T)^{1/3} \left(\frac{L}{\gamma} + \frac{L_g}{\gamma_g} \right)^{2/3}$$

Proof. The algorithm under consideration here is as follows:

1. For the first KN rounds, pick each arm N times (with N to be specified later)

2. For $t > NK$, for each arm a, set $\mu_a^t = \hat{\mu}_a$ where $\hat{\mu}_a$ is the empirical mean reward of arm a from the explore rounds and set $\pi^t(a) = \frac{g(\vec{\mu^t_{G(a)}})}{\sum_{G' \in \mathcal{G}} g(\vec{\mu^t_{G'}})} \cdot \frac{f(\mu_a^t)}{\sum_{a' \in G(a)} f(\mu_{a'}^t)}$

Using Hoeffding's inequality, for a fixed arm a, w.p. at least $1 - \frac{1}{T^4}$,

$$|\mu_a^* - \hat{\mu}_a| \leq \sqrt{\frac{2 \log T}{N}}$$

Let $\mathcal{E}$ be the event that for all arms a, $|\mu_a^* - \hat{\mu}_a| \leq \sqrt{\frac{2 \log T}{N}}$. Using union bound, $P[\mathcal{E}] \geq 1 - \frac{1}{T^3}$ (assuming $K \leq T$). Let's assume that event $\mathcal{E}$ is true. Recall that the fairness regret is given by

$$\text{FR}_{\mathcal{A}}^T = \sum_t^T \left(\sum_{G \in \mathcal{G}} \left(|\Pi(G, \mu^t) - \Pi(G, \mu^*)| + \Pi(G, \mu^*) \cdot \sum_{a \in G} |\Gamma(a, \mu^t) - \Gamma(a, \mu^*)| \right) \right)$$

Consider the term, $\sum_t^T \left(\sum_{G \in \mathcal{G}} \left(\Pi(G, \mu^*) \cdot \sum_{a \in G} |\Gamma(a, \mu^t) - \Gamma(a, \mu^*)| \right) \right)$. Using the same steps as in Wang *et al.* [21][Theorem 3.2.2], for a fixed group G and $t > KN$, we can show that,

$$\begin{aligned}
\sum_{a \in G} |\Gamma(a, \mu^t) - \Gamma(a, \mu^*)| &\leq \sum_{a \in G} \frac{|\mu_a^t - \mu_a^*| L\Gamma(a, \mu^*)}{\gamma} \\
&= \sum_{a \in G} \frac{|\hat{\mu}_a - \mu_a^*| L\Gamma(a, \mu^*)}{\gamma} \\
&\leq \sum_{a \in G} \frac{L\Gamma(a, \mu^*)}{\gamma} \sqrt{\frac{2 \log T}{N}}
\end{aligned}$$

Summing over all groups and rounds,

$$\begin{aligned}
&\sum_{t}^{T}\sum_{G\in\mathcal{G}}\sum_{a\in G}\Pi(G,\mu^*)\cdot|\Gamma(a,\mu^t)-\Gamma(a,\mu^*)| \\
&\leq 2KN+\sum_{t=2KN+1}^{T}\sum_{G\in\mathcal{G}}\sum_{a\in G}\frac{L\Pi(G,\mu^*)\Gamma(a,\mu^*)}{\gamma}\sqrt{\frac{2\log T}{N}} \\
&= 2KN+\sum_{t=KN+1}^{T}\sum_{a\in[K]}\frac{L\pi^*(a)}{\gamma}\sqrt{\frac{2\log T}{N}} \\
&\leq 2KN+\frac{TL}{\gamma}\sqrt{\frac{2\log T}{N}}
\end{aligned}$$

Similarly, we can show that for $t > KN$,

$$\sum_{G\in\mathcal{G}}\left(|\Pi(G,\mu^t)-\Pi(G,\mu^*)|\right)\leq\sum_{G\in\mathcal{G}}\frac{|\mu_G^t-\mu_G^*|L_g\Pi(G,\mu_G^*)}{\gamma_g}=\sum_{G\in\mathcal{G}}\frac{|\hat{\mu}_G-\mu_G^*|L_g\Pi(G,\mu_G^*)}{\gamma_g} \tag{7.2}$$

where $\hat{\mu}_G=\frac{\sum_{a\in G}\hat{\mu}_a}{|G|}$. Considering $|\hat{\mu}_G-\mu_G^*|$, we have

$$|\hat{\mu}_G-\mu_G^*|=\left|\frac{\sum_{a\in G}\hat{\mu}_a}{|G|}-\frac{\sum_{a\in G}\mu_a^*}{|G|}\right|\leq\frac{\sum_{a\in G}|\hat{\mu}_a-\mu_a^*|}{|G|}\leq\sqrt{\frac{2\log T}{N}}$$

Summing over all rounds t and groups we get

$$\begin{aligned}
\sum_{t}^{T}\sum_{G\in\mathcal{G}}\left(|\Pi(G,\mu^t)-\Pi(G,\mu^*)|\right)&\leq 2KN+\sum_{t=KN+1}^{T}\sum_{G\in\mathcal{G}}\frac{|\hat{\mu}_G-\mu_G^*|L_g\Pi(G,\mu_G^*)}{\gamma_g} \\
&\leq 2KN+\frac{TL_g}{\gamma_g}\sqrt{\frac{2\log T}{N}}
\end{aligned}$$

Adding both the terms, with probability at least $1-\frac{1}{T^3}$,

$$\begin{aligned}
\text{FR}_{\mathcal{A}}^T&\leq 4KN+\frac{TL}{\gamma}\sqrt{\frac{2\log T}{N}}+\frac{TL_g}{\gamma_g}\sqrt{\frac{2\log T}{N}} \\
&\leq 4T^{2/3}(K\log T)^{1/3}\left(\frac{L}{\gamma}+\frac{L_g}{\gamma_g}\right)^{2/3}
\end{aligned}$$

The upper bound is obtained by setting $N=\left(\frac{T}{K}\right)^{2/3}(\log T)^{1/3}\left(\frac{L}{\gamma}+\frac{L_g}{\gamma_g}\right)^{2/3}$.

□

Using similar ideas, we can show an upper bound for the reward regret of Algorithm 15 as well. We need one one small result before that, that shows that if you points are close then their multiples are close as well.

Lemma 7.2.6. *Let $a, b, a', b' > 0$ such that $|a - a'| \leq \delta_a$, and $|b - b'| \leq \delta_b$, then*

$$|ab - a'b'| \leq |a|\delta_b + |b|\delta_a + \delta_a\delta_b$$

Theorem 7.2.7. *If f is L-Lipschitz, lower bounded by γ and g is a scalar function of the average mean rewards, is L_g-Lipschitz and lower bounded by γ_g, using an explore then exploit algorithm (Algorithm 15) with fixed exploration of $N = \tilde{O}(T^{2/3})$, the reward regret satisfies the following upper-bound:*

$$RR_{\mathcal{A}}^T \leq 4T^{2/3}(K \log T)^{1/3} \left(\frac{L}{\gamma} + \frac{L_g}{\gamma_g}\right)^{2/3}$$

Proof.

$$\begin{aligned}
\text{RR}_{\mathcal{A}}^T &= \sum_{t=1}^{T} \sum_{a \in [K]} \pi^*(a)\mu_a^* - \sum_{t=1}^{T} \sum_{a \in [K]} \pi^t(a)\mu_a^* \\
&\leq \sum_{t=1}^{T} \sum_{a \in [K]} |\pi^*(a) - \pi^t(a)|\mu_a^* \\
&\leq \sum_{t=1}^{T} \sum_{a \in [K]} |\pi^*(a) - \pi^t(a)|
\end{aligned}$$

Using Hoeffding's inequality, for a fixed arm a, w.p. at least $1 - \frac{1}{T^4}$,

$$|\mu_a^* - \hat{\mu}_a| \leq \sqrt{\frac{2 \log T}{N}}$$

Let $\mathcal{E}$ be the event that for all arms a, $|\mu_a^* - \hat{\mu}_a| \leq \sqrt{\frac{2 \log T}{N}}$. Using union bound, $P[\mathcal{E}] \geq 1 - \frac{1}{T^3}$ (assuming $K \leq T$). Let's assume that event $\mathcal{E}$ is true. For a fixed group G,

let $|\Pi(G,\mu^*) - \Pi(G,\hat{\mu})| \leq \Delta_G$ and for a fixed arm a, let $|\Gamma(a,\mu^*) - \Gamma(a,\hat{\mu})| \leq \epsilon_a$

Consider a fixed arm a and round $t > TK$,

$$
\begin{aligned}
|\pi^*(a) - \pi^t(a)| &= |\Pi(G(a),\mu^*)\Gamma(a,\mu^*) - \Pi(G(a),\hat{\mu})\Gamma(a,\hat{\mu})| \\
&\overset{Lemma\ 7.2.6}{\leq} \Pi(G(a),\mu^*)\epsilon_a + \Gamma(a,\mu^*)\Delta_{G(a)} + \Delta_{G(a)}\epsilon_a \\
&\leq \Pi(G(a),\mu^*)\epsilon_a + \Delta_{G(a)} + \Delta_{G(a)}\epsilon_a
\end{aligned}
$$

Summing over all arms a ,

$$
\begin{aligned}
&\sum_{a\in[K]} |\pi^*(a) - \pi^t(a)| \\
&\leq \sum_{a\in[K]} (\Pi(G(a),\mu^*)\epsilon_a + \Delta_{G(a)} + \Delta_{G(a)}\epsilon_a) \\
&\leq \sum_{a\in[K]} \frac{L\Pi(G(a),\mu^*)\Gamma(a,\mu^*)}{\gamma}\sqrt{\frac{2\log T}{N}} + \frac{L_g}{\gamma_g}\sqrt{\frac{2\log T}{N}}(1 + \frac{L}{\gamma}\sqrt{\frac{2\log T}{N}}) \\
&\leq \frac{L}{\gamma}\sqrt{\frac{2\log T}{N}} + \frac{L_g}{\gamma_g}\sqrt{\frac{2\log T}{N}}\left(1 + \frac{L}{\gamma}\sqrt{\frac{2\log T}{N}}\right)
\end{aligned}
$$

Summing over T, we get

$$
\begin{aligned}
RR_{\mathcal{A}}^T &\leq 2KN + T\left(\frac{L}{\gamma}\sqrt{\frac{2\log T}{N}} + \frac{L_g}{\gamma_g}\sqrt{\frac{2\log T}{N}}\left(1 + \frac{L}{\gamma}\sqrt{\frac{2\log T}{N}}\right)\right) \\
&\leq 4T^{2/3}(K\log T)^{1/3}\left(\frac{L}{\gamma} + \frac{L_g}{\gamma_g}\right)^{2/3}
\end{aligned}
$$

The upper bound is obtained by setting $N = \left(\frac{T}{K}\right)^{2/3}(\log T)^{1/3}\left(\frac{L}{\gamma} + \frac{L_g}{\gamma_g}\right)^{2/3}$. $\square$

Appendices

APPENDIX A

MISSING PROOFS AND ADDITIONAL EXPERIMENTS FROM CHAPTER 3

A.1 Missing Proofs

We now prove the main result about the vulnerability of mean based algorithms (Theorem 3.4.1). That is, for any mean based bandit algorithm that achieves sub-linear regret in the absence of data-corruptions, there always exists an instance where an adversarial data corruption attack with $o(T)$ corruption level can make the algorithm suffer linear regret $\textsc{Reg}_{\mathcal{A}}(T) = \Omega(T)$ in expectation.

A.1.1 Proof for Theorem 3.4.1

Proof. Denote the two arms in instances with two arms as a_1 and a_2. Given an instance where the means of both arms are 0. For any constant C_1, there is always at least one arm such that it gets at least $C_1/2$ picks with probability at least $1/2$, denote such arm as a_1. We consider an instance (1) where a_1 is the optimal arm:

$$\mu_{a_2}^{(1)} = \mu,$$

$$\mu_{a_1}^{(1)} > \mu_{a_2}^{(1)}.$$

We will perform the observation free attack on instance (1). In the first phase of attack, the rewards are always 0 for any arm. By the end of the first case, for instance (1), from the way we set a_1, we have with probability at least $1/2$, the following will happen:

$$n_{a_1}^{(1)} \geq \frac{C_1}{2}, \hat{\mu}_{a_1}^{(1)} = 0,$$

$$n_{a_2}^{(1)} \leq \frac{C_1}{2}, \hat{\mu}_{a_2}^{(1)} = 0.$$

Let G_1 be the event that the above is true in instance (1), we know that $P\{G_1\} \geq 1/2$.

Next, consider another instance (2) where the mean reward of a_2 is 1, and the mean reward of other arm is 0:

$$\mu_{a_1}^{(2)} = 0,$$

$$\mu_{a_2}^{(2)} = 1.$$

For instance (2), we corrupt the first C_1 rounds and set the rewards to be 0 for all arms, then stop corruption. Let $N_1^{(2)}$ be the number of rounds when the algorithm pick arm 1 after the corruption ends. Let f_1 be the value such that $P\{N_1^{(2)} \geq f_1\} = 1/2$. The expected regret of the algorithm is at least $R^{(2)}(T) \geq 1/2 f_1$. So $f_1 \leq 2R^{(2)}(T)$, which has to be sublinear or otherwise the algorithm has linear expected regret in instance (2).

Next we focus on the second phase of attack in instance (1). Let $C_2 = f_1 + \alpha C_1$ where α is a parameter to be specified later. Up the end of this phase, what happened in (1) is the same as that in (2). So with probability $1/2$, a_1 is picked for less than f_1 rounds in this phase. Denote such Event as G_2, then $P\{G_2\} = 1/2$. If both G_1 and G_2 are true, by the end of the second phase of attack, the following is true :

$$n_{a_1}^{(1)} \geq \frac{C_1}{2}, \hat{\mu}_{a_1}^{(1)} = 0,$$

$$n_{a_2}^{(1)} \geq \alpha C_1, \hat{\mu}_{a_2}^{(1)} \geq \frac{2\alpha}{2\alpha + 1}.$$

Next we focus on the last phase of attack in instance (1) where the corruption is ended. For any value of n, if a_2 get picked for n times in this phase, then by Hoeffding inequality inequality, with probability at least $1 - 1/T$, the reward from these n rounds is at least $\mu n - \sqrt{\log(T)n}$ for any $n \leq T$. Set $\alpha = \frac{\frac{\log(T)}{2\mu C_1} + \frac{\mu}{4}}{1-\mu/2}$, the corresponding empirical mean of a_2

satisfies

$$\bar{\mu} = \frac{C_1 \cdot \alpha + n\mu - \sqrt{n \log(T)}}{C_1(\alpha + 1/2) + n} \geq \mu/2.$$

That is, in the last phase, with probability at least $1-1/T$, the empirical mean of a_2 is always greater than $\mu/2$. Let G_3 denote the event where the above happens, so $P\{G_3\} \geq 1 - 1/T$.

Before proceeding, we introduce an instance (3) where the reward of arm a_1 is always $\mu/4$ and the reward of a_2 is always $\mu/2$. Let n_1^t and n_2^t be the number of rounds a_1 and a_2 get selected by round t. Define random variables $\{Y_1, \ldots, Y_{T/2}\}$ where Y_n is n_1^t if exists a t such that $n_2^t = n$, and $T - n$ if such t doesn't exists. It is clear that $P\{Y_n < 0\} = 0$, $P\{Y_n < T\} = 1$, and $P\{Y_n < x\} \leq P\{Y_n < x\} + 1$. So we could always find an integer k such that $P\{Y_{T/2} < k\} = 1/2$, and such k must be sublinear in T or otherwise the regret in instance (3) will be linear. Y_n also satisfies $Y_n \in [Y_{n-1}, Y_{n-1} + 1, \ldots, T - n]$, and $P\{Y_n = Y_{n-1} + i | Y_{n-1} = y_{n-1}\} \geq P\{Y_n = Y_{n-1} + j | Y_{n-1} = y_{n-1}\}$ for all $0 \leq i \leq j$ and y_{n-1}. The purpose of introducing instance (3) is to show that if the algorithm have sublinear regret in this instance, then with probability $1/2$, it won't pick a_1 for more than k times. Then in stance (1), by choosing big enough C_1 and C_2, with probability at least $1/2$, it won't pick a_1 for more than k times, so the algorithm will have linear regret in instance (1).

Now back to instance (1) and set $C_1 = (8/\mu - 2)k$, so at the beginning of the last phase, a_1 has already been picked for at least $(4/\mu - 1)k$ rounds. Then the empirical mean of a_1 will not exceed $\mu/4$ before it get at least k picks from this phase. Then by the definition of mean based algorithm, we know that before a_1 get its kth pick, the probability a_1 get picked in instance (1) is always less than that in instance (3) for the same number of rounds a_2 get picked. Let n_1^t and n_2^t as the number of rounds arm a_1 and a_2 get picked in the last phase by round t. Define random variables $\{Z_1, \ldots, Z_{T/2}\}$ in the same way as Y_n where $Z^n = n_1^t$ if exists t such that $n_2^t = n$ and $Z^n = T - n$ if such t doesn't exists. Z_n also satisfies $Z_n \in [Z_{n-1}, Z_{n-1} + 1, \ldots, T - n]$, and $P\{Z_n = Z_{n-1} + i | Z_{n-1} = z_{n-1}\} \geq P\{Z_n = Z_{n-1} + j | Z_{n-1} = z_{n-1}\}$ for all $0 \leq i \leq j$ and z_{n-1}. The relation between Z_n and Y_n satisfies: $P\{Z_n = x | Z_{n-1} = x\} \geq P\{Y_n = x | Y_{n-1} = x\}$ for all $x \leq k$ and

$P\{Z_n = x+i|Z_{n-1} = x\} \leq P\{Y_n = x+i|Y_{n-1} = x\}$ for all $i > 0$ and $x+i \leq k$. Intuitively, Z_n "grows" slower than Y_n before it exceeds k, so Y_n is more likely to reach k than Z_n. Next are we going to strictly prove that $P\{Y_{T/2} \leq k\} \leq P\{Z_{T/2} \leq k\}$.

Note that $P\{Y_n \leq k\}$ depends on $P\{Y_m|Y_{m-1}\}$ for all $m \leq n$. The idea of the proof is to show that by substituting each $P\{Y_m|Y_{m-1}\}$ by $P\{Z_m|Z_{m-1}\}$, the probability of $P\{Y_n \leq k\}$ will increase. We introduce another series of random variables $\{F_1^1, \ldots, F_{T/2}^1\}$ where $\{F_n^1\}$ is almost the same as $\{Y_n\}$ except that $P\{F_m^1|F_{m-1}^1\} = P\{Z_m|Z_{m-1}\}$ for a specific m. We want to show that $P\{Y_{T/2} \leq k\} \leq P\{F_{T/2}^1 \leq k\}$. After that, we can construct $\{F_n^2\}$ which is almost the same as $\{F_n^1\}$ except for $P\{F_{m'}^2|F_{m'-1}^1\} = P\{Z_{m'}|Z_{m'-1}\}$ where $m' \neq m$. For the same reason we will have $P\{F_{T/2}^1 \leq k\} \leq P\{F_{T/2}^2 \leq k\}$. Repeat this process until $\{F_n^{T/2}\}$ which is the same as $\{Z_n\}$, then we have $P\{Y_{T/2} \leq k\} \leq P\{F_{T/2}^1 \leq k\} \leq P\{F_{T/2}^2 \leq k\} \leq \ldots \leq P\{F_{T/2}^{T/2} \leq k\} = P\{Z_{T/2} \leq k\}$. Next we will prove that $P\{Y_{T/2} \leq k\} \leq P\{F_{T/2}^1 \leq k\}$.

First, we can write $P\{Y_{T/2} \leq k\}$ as

$$
\begin{aligned}
&P\{Y_{T/2} \leq k\} \\
&= \sum_{x=0}^{k} P\{Y_{T/2} \leq k|Y_{m-1} = x\} \cdot P\{Y_{m-1} = x\} \\
&= \sum_{x=0}^{k} P\{Y_{m-1} = x\} \cdot \sum_{y=x}^{k} P\{Y_n \leq k|Y_m = y, Y_{m-1} = x\} \cdot P\{Y_m = y|Y_{m-1} = x\} \\
&= \sum_{x=0}^{k} P\{F_{m-1}^1 = x\} \cdot \sum_{y=x}^{k} P\{F_n^1 \leq k|F_m^1 = y\} \cdot P\{Y_m = y|Y_{m-1} = x\}
\end{aligned}
$$

The difference between $P\{Y_{T/2} \leq k\}$ and $P\{F^1_{T/2} \leq k\}$ can be written as

$$
\begin{aligned}
&P\{Y_{T/2} \leq k\} - P\{F^1_{T/2} \leq k\} \\
=&\sum_{x=0}^{k} P\{F^1_{m-1} = x\} \cdot \sum_{y=x}^{k} P\{F^1_n \leq k | F^1_m = y\} \cdot \\
&(P\{Y_m = y | Y_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\})
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{y=x}^{k} P\{F^1_n \leq k | F^1_m = y\} \cdot (P\{Y_m = y | Y_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\}) \\
=&P\{Y_n \leq k | Y_m = y\} \cdot (P\{Y_m = x | Y_{m-1} = x\} - P\{F^1_m = x | F^1_{m-1} = x\}) \\
&+ \sum_{y=x+1}^{k} P\{F^1_n \leq k | F^1_m = y\} \cdot (P\{Y_m = y | Y_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\}) \\
=&P\{Y_n \leq k | Y_m = y\} \cdot \sum_{z=x+1}^{T-m} (P\{F^1_m = z | F^1_{m-1} = x\} - P\{Y_m = z | Y_{m-1} = x\}) \\
&+ \sum_{y=x+1}^{k} P\{F^1_n \leq k | F^1_m = y\} \cdot (P\{Y_m = y | Y_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\}) \\
\leq&P\{Y_n \leq k | Y_m = y\} \cdot \sum_{z=x+1}^{y} (P\{F^1_m = z | F^1_{m-1} = x\} - P\{Y_m = z | Y_{m-1} = x\}) \\
&+ \sum_{y=x+1}^{k} P\{F^1_n \leq k | F^1_m = y\} \cdot (P\{Y_m = y | Y_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\}) \\
=&\sum_{y=x+1}^{k} (P\{F^1_n \leq k | F^1_m = x\} - P\{F^1_n \leq k | F^1_m = y\}) P\{F^1_m = y | F^1_{m-1} = x\} \\
&- \sum_{y=x+1}^{k} (P\{F^1_n \leq k | F^1_m = x\} - P\{F^1_n \leq k | F^1_m = y\})(P\{Y_m = y | Y_{m-1} = x\})
\end{aligned}
$$

We can directly have $P\{F^1_m = y | F^1_{m-1} = x\} - P\{F^1_m = y | F^1_{m-1} = x\} \leq 0$, for the other term, we have:

$$
\begin{aligned}
&P\{F_n^1 \le k | F_m^1 = x\} \\
&= P\{F_n^1 \le k | F_{m+1}^1 \le k, F_m^1 = x\} \cdot P\{F_{m+1}^1 \le k | F_m^1 = x\} \\
&= P\{F_n^1 \le k | F_{m+1}^1 \le k\} \cdot P\{F_{m+1}^1 \le k | F_m^1 = x\} \\
&\ge P\{F_n^1 \le k | F_{m+1}^1 \le k\} \cdot P\{F_{m+1}^1 \le k | F_m^1 = y\} \\
&= P\{F_n^1 \le k | F_m^1 = y\}
\end{aligned}
$$

So eventually we have

$$
\begin{aligned}
&P\{Y_{T/2} \le k\} - P\{F_{T/2}^1 \le k\} \\
&= \sum_{x=0}^{k} P\{F_{m-1}^1 = x\} \cdot \sum_{y=x}^{k} P\{F_n^1 \le k | F_m^1 = y\} \cdot (P\{Y_m = y | Y_{m-1} = x\} \\
&- P\{F_m^1 = y | F_{m-1}^1 = x\}) \\
&\le \sum_{x=0}^{k} P\{F_{m-1}^1 = x\} \cdot \sum_{y=x+1}^{k} (P\{F_n^1 \le k | F_m^1 = x\} \\
&- P\{F_n^1 \le k | F_m^1 = y\})(P\{F_m^1 = y | F_{m-1}^1 = x\} - P\{Y_m = y | Y_{m-1} = x\}) \\
&\le 0
\end{aligned}
$$

As discussed before, by the same process we have $P\{F_{T/2}^1 \le k\} - P\{F_{T/2}^2 \le k\} \le 0$ and so on. So $P\{Y_{T/2} \le k\} \le P\{F_{T/2}^1 \le k\} \le P\{F_{T/2}^2 \le k\} \le \ldots \le P\{F_{T/2}^{T/2} \le k\} = P\{Z_{T/2} \le k\}$. Next we will prove that $P\{Y_{T/2} \le k\} \le P\{F_{T/2}^1 \le k\}$. That is, with probability at least $1/2$, in instance (1), a_2 will be picked for more than $T/2$ rounds and by that time a_1 is picked for less than k rounds.

Suppose the algorithm guarantee sublinear regret in instances (2) and (3). Let $\mu = 1/2$ and the mean reward of the optimal arm as 1, set $C_1 = 14k$ and $C_2 = f_1 + \frac{3}{4}\log(T) + \frac{7}{3}k$, the expected regret for the algorithm in instance (1) is at least $T/16$. $\square$

A.1.2 Proof for Theorem 3.5.1

Proof. Let 1 as the index of the target arm. Under the adversarial attack, in the first phase of attack when $t \leq C_1$, the empirical mean of any arm will always be 0, so the empirical upper confidence for each arm j satisfies

$$\mathsf{UCB}_j^t = \hat{\mu}_j^t + \sqrt{\frac{\log T}{n_j^t}} = \sqrt{\frac{\log T}{n_j^t}}.$$

It is clear that $\text{argmax}_j \, \mathsf{UCB}_j^t = \text{argmin}_j \, n_j^t$. So an arm could get its $n+1^{th}$ pick only after all other arms get selected at least n times. That is, arms will be selected in turn. Hence, when $t = C_1 + 1$, all arms will be selected for C_1/K times.

In the second phase of attack When $C_1 < t \leq C_1 + C_2$, the empirical mean of the target arm is increasing whenever it get selected while that of the others remain 0. If we choose $C_1 \geq \frac{4 \log T}{K}$, then the upper confidence bound of the target arm 1 when it gets n picks at this period satisfies:

$$\begin{aligned}
\mathsf{UCB}_1^t &= \hat{\mu}_1^t + \sqrt{\frac{\log T}{n_1^t}} \\
&= \frac{n}{n + 4 \log T / K^2} + \sqrt{\frac{\log T}{C_1/K + n}} \\
&\geq \sqrt{\frac{\log T}{C_1/K}} = \mathsf{UCB}_{i \neq 1}^t
\end{aligned}$$

So the target arm will get all the C_2 picks at this period. We choose $C_1 = \max\{\frac{K \log(T)}{\mu_1^2}, \frac{4 \log T}{K}\}$, so that the upper bound of other arms at the end of the second phase will be no greater than μ_1. Considering the fact that $K \geq 2$ and $\mu_1 \leq 1$, we have $C_1 = \frac{K \log(T)}{\mu_1^2}$. Then we choose $C_2 = \frac{\mu_1}{1-\mu_1} C_1$ so that at the end of the second phase, the empirical mean of the second arm is its true mean μ_1

In the last phase of attack when $t > C_1 + C_2$, we will show that the target arm will be picked for all rounds with a high probability. When the target arm get n picks in this phase,

by Hoeffding inequality, with probability at least $1 - 1/T$, the total reward generated from these n rounds is greater than $\mu_1 n - \sqrt{n \log T}$ for any value of $n < T$. Denote the number of rounds the target arm get picked before $t = C_1 + C_2$ as m, then the upper bound of the target arm satisfies

$$\mathsf{UCB}_j^t = \hat{\mu}_1^t + \sqrt{\frac{\log T}{n_1^t}} \geq \mu_1 - \frac{\sqrt{n \log T}}{n+m} + \sqrt{\frac{\log T}{n+m}} > \mu_1.$$

Therefore the target arm's upper confidence bound is always the highest no matter how many times it get picks in the last phase, which means it will always get picked with probability at least $1 - 1/T$.

In conclusion, to defeat UCB algorithm, the observation free attack corrupt the first $\max\{\frac{K \log(T)}{\mu_1^2}, \frac{4 \log T}{K}\}/(1 - \mu_1)$ rounds, and the number of rounds arm other than the target get selected is less than $\frac{(K-1)\log(T)}{\mu_1^2}$ with probability at least $1 - \frac{1}{T}$. □

A.1.3 Proof for Theorem 3.5.2

Proof. We refer to the rounds where the algorithm randomly pick an arm from all arms as "explore" rounds. Under the corruption from adversary algorithm, in the first phase of attack when $t < C_1$, all arms have the same probability to get picked because their empirical means are all 0. So each arm will get picked no less than

$$n_1 = C_1/K - \sqrt{C_1 \log T}$$

rounds and no more than

$$n_2 = C_1/K + \sqrt{C_1 \log T}$$

with probability at least $1 - \frac{K}{T}$ given by Hoeffding inequality. Next we will discuss the case where the above is true.

In the second phase of attack when $C_1 < t \leq C_1 + C_2$, once the target arm get one pick,

its empirical mean will be the highest, and it will be selected with probability at least $1-\epsilon$. With probability at least $1-1/T$, the target arm will get its first pick after $K\log(T)$ rounds. After that, with probability at least $1-1/T$, the target arm will get picked for at least

$$n(C_2) = (C_2 - K\log(T))(1-\epsilon) - \sqrt{C_2\log(T)}$$

times. Denote μ as the empirical mean of the target arm, to simplify the analysis, we choose C_2 big enough such that the target arm can get picked at least $n_3 = \max\{\frac{\log T}{\mu^2}, n_2\frac{\mu}{1-\mu}\}$ times during this period. The reason we choose this n_3 is to make sure that the empirical mean of target arm is high enough when $t > C_1 + C_2$, which will be shown later. To make sure $n(C_2) \geq n_3$, we can choose

$$C_2 = K\log T + \frac{2n_3}{1-\epsilon}$$

.

In the last phase of attack when $t > C_1 + C_2$, we want to find a lower bound on empirical mean of the target arm. Note that $n_3 \geq \frac{4\log T}{\mu^2}$, so that empirical mean of the target arm at the beginning of this phase $t = C_1 + C_2 + 1$ is greater than μ. Denote the number of rounds the target arm get picked after $t = C_1 + C_2$ as m, the empirical mean of the target arm satisfies:

$$\begin{aligned}
\hat{\mu} &\geq \frac{\mu n_3 + \mu m - \sqrt{m\log T}}{n_3 + m} \\
&= \mu - \frac{\sqrt{m\log T}}{n_3 + m} \\
&\geq \mu - 0.5\sqrt{\frac{\log T}{n_3}} \\
&= 0.5\mu
\end{aligned}$$

Therefore, before an arm other than the target arm has its empirical mean greater than 0.5μ, the probability it get picked is ϵ/K. We want C_1 to be big enough such that the empirical means of other arm are always less than $\mu/2$ in the last phase. From Hoeffding

inequality, with probability at least $1 - 1/T$, an arm will get picked from explore rounds for at most $T \log T\epsilon/K$ rounds. If the arm never get picked from the exploit rounds, its empirical mean satisfies:

$$\hat{\mu}_i \leq \frac{T \log T\epsilon/K}{T \log T\epsilon/K + n_1}.$$

Set

$$C_1 = T \log T\epsilon(4/\mu - 2),$$

such that

$$n_1 = (T \log T\epsilon/K)(2/\mu - 1),$$

then we have $\hat{\mu}_i \leq \mu/2$. So with this C_1, with probability at least $1 - K/T$, the empirical mean of other arms never exceed that of the target arm hence get not picks from the explore rounds. Based on such C_1, the corresponding C_2 is

$$C_2 = K \log T + \frac{2}{1-\epsilon}(\max\{\frac{\log T}{\mu^2}, \frac{\mu}{1-\mu}(C_1/K + \sqrt{C_1 \log T})\})$$

With such C_1 and C_2, the ϵ-greedy algorithm will pick arms other than the target arm by at most $C_1 + T\epsilon + \sqrt{C_2 \log T}$ times with probability at least $1 - (2K+2)/T$.

□

A.1.4 Proof for Theorem 3.5.3

Proof. Let 1 be the index of the target arm. When $t < C_1$, we want to show that all arms will get picked for around C_1/K rounds. Let's start with the case where $K = 2$. Denote Δ^t as the difference of number of rounds the other get picked, and $\Delta^{t+1} - \Delta^t$ as δ^t. The probability that the arm which get more picked before get picked this round is no greater than $1/2$. That is, if $\Delta^t \geq 0$, $P\{\delta^t = 1\} \leq 1/2$ and $P\{\delta^t = -1\} \geq 1/2$; if $\Delta^t \leq 0$, $P\{\delta^t = 1\} \geq 1/2$ and $P\{\delta^t = -1\} \leq 1/2$. Since $\Delta^{t=C_1+1} = \sum_{t=1}^{C_1} \delta^t$, with probability at least $1 - 1/T$, $\Delta^{t=C_1+1} \leq \sqrt{C_1 \log T}$. In the case where $K > 2$, we can define $\Delta^t_{i,j}$ and $\delta^t_{i,j}$ as the Δ^t and $\delta^t_{i,j}$ arm i and j, and by similar argument we have with probability at least

$1 - 1/T$, $\Delta_{i,j}^{t=C_1+1} \leq \sqrt{C_1 \log T}$. This means at round $t = C_1 + 1$, with probability at least $1 - K/T$, the number of rounds any arm get picked is no less than

$$n_1 = \frac{C_1 - (K-1)\sqrt{C_1 \log T}}{K}$$

, and no greater than

$$n_2 = \frac{C_1 + (K-1)\sqrt{C_1 \log T}}{K}.$$

When $C_1 < t \leq C_1 + C_2$, denote X^j as the number of rounds between the target arm get its $(j-1)^{th}$ and j^{th} pick. After the target arm get its $(j-1)^{th}$ pick before its jth pick, in the worst case, its beta distribution is $B(j, 1+n_2)$, and that of any other arm is $B(1, 1+n_1)$. By simple arithmetic calculation, we have when $j = 1$, $P\{\theta_1 < \theta_i\} = \frac{\beta}{1+\beta}$, and when $j \geq 2$, $P\{\theta_1 < \theta_i\} \leq \frac{1}{j\beta}$ where $\beta = \frac{n_1+1}{n_2+1}$, so $P\{\theta_1 > \theta_{i\neq 1}\} \geq (1 - \frac{1}{j\beta})^{K-1}$. When $j = 1$, we have $P\{\theta_1 > \theta_{i\neq 1}\} \geq (\frac{\beta}{1+\beta})^{K-1}$. The probability that the target arm be selected is at least $1/2^{K-1}$, When $j < \frac{1}{\beta(1-2^{1-K})} := n_3$, and at least $1/2^{K-1}$. when $j \geq n_3$. With probability at least $1 - 1/T$, the target arm will be picked for at least $(C_2 - n_3(\frac{\beta}{1+\beta})^{1-K} \log T)/2 - \sqrt{C_2 \log T}$ rounds.

We select C_1 and C_2 to be large enough such that with high probability, when $t > C_1 + C_2$, $\theta_1 > \mu/2$ and $\theta_{i\neq 1} < \mu/2$, so that the target arm will get all the picks. We set $C_1 = \frac{4 \log T}{\mu^2}$, and $C_2 = n_3(\frac{\beta}{1+\beta})^{1-K} \log T + 2\frac{\mu}{1-\mu}C_1$, then by $t = C_1 + C_2$, arms other than the target arm is picked for at least n_1 times, and the target arm's is picked for at least n_2 times with mean no less than μ. By result from [77], this can ensure that with probability at least $1 - K/T$, $\theta_{i\neq 1}^t < \mu/2$ and $\theta_1^t > \mu/2$ true for all rounds. So with probability at least $1 - (2K+1)/T$, the target arm will get all picks when $t > C_1 + C_2$, with C_1 and C_2 as given above. □

A.2 Chernoff Bounds

Lemma A.2.1 (Chernoff Bounds)**.** *Let $X_1, \ldots, X_n$ be independent random variables, and X_i lies in the interval $[0, 1]$. Define $X = \sum_{i=1}^n X_i$ and denote $E[X] = \mu$. For any $\delta \in [0, 1]$,*

we have ***Chernoff lower tail:***

$$Pr\{X < (1-\delta)\mu\} \leq \exp(-\frac{\mu\delta^2}{3})$$

and we have ***Chernoff upper tail:***

$$Pr\{X > (1+\delta)\mu\} \leq \begin{cases} \exp(-\frac{\mu\delta}{3}) & \textbf{for } \delta > 1 \\ \exp(-\frac{\mu\delta^2}{3}) & \textbf{for } \delta \in [0,1] \end{cases}$$

The proofs for the inequalities in Lemma A.2.1 can be found in Theorem 4.4 and Theorem 4.5 of [146]

A.3 Additional Experiments

Here we run both attack methods with or without knowing the mean reward μ of the target arm against UCB, Thompson sampling, and ϵ-greedy bandit algorithms in different instances, where ϵ is set to be $T^{2/3}$ in ϵ-greedy algorithm. For each pair of attack method and bandit algorithm, we run the experiments in three instances where there are two arms, and the mean reward for the optimal arm is always 1 while the mean reward for the target arm is $\mu = 0.3, 0.5, 0.7$ respectively. First we verify that our main attack algorithm 4 indeed manipulates the behavior of the bandit algorithms as the theory suggests. The parameters for this attack method is given by theorem 3.5.1 when attacking UCB algorithm, theorem 3.5.2 when attacking ϵ-greedy algorithm, and theorem 3.5.3 when attacking Thompson sampling algorithm. In figure A.1, for this attack method, we plot the number of rounds n when the non-target arm get selected versus the total number of rounds T for UCB algorithm in subfigure (a1), Thompson sampling algorithm in subfigure (a2), and ϵ-greedy algorithm in subfigure (a3). The plots show that there is a linear dependence between n and $\log(T)$ in (a1) and (b1), and between n and $T^{2/3}$ in (c1), which agrees with our theoretical guarantee. Each experiment is repeated for 100 times.

Next we show that the modified attack which needs to estimate μ can also manipulate the algorithms without using a high corruption budget. In figure A.2, in subfigures a1), b1) and c1), we plot the number of corruption rounds needed by the algorithm vs the total number of rounds T in the case when the algorithm doesn't know the true mean μ for the bandit algorithms UCB, Thompson Sampling, and ϵ-Greedy respectively. In subfigures a2), b2) and c2), we plot the corresponding number of times the non target arm was pulled for the corresponding corruption levels in the plots a1), b1) and c1) respectively. The plots show that even when the algorithm doesn't the mean reward, there is still a linear dependence between the corruption level C and $\log(T)$ in (a1) and (b1), and between C and $T^{2/3}$ in (c1), and similarly a linear dependence between the number of times the non-target arm is pulled n and $\log(T)$ in (a2) and (b2), and between n and $T^{2/3}$ in (c2). These results show that, along with strong theoretical guarantees, our attack methodologically also perform well empirically.

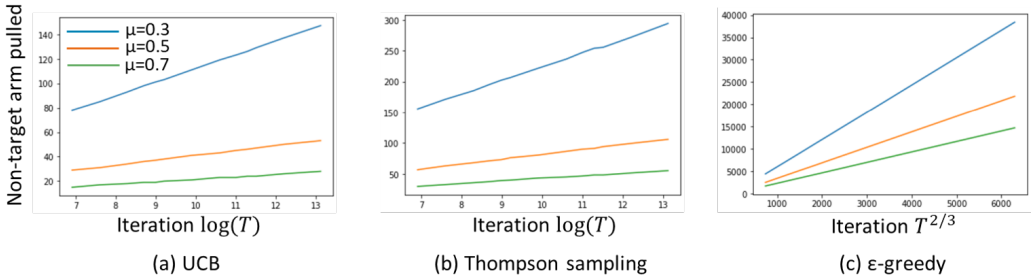


Figure A.1: The attack which knows the mean reward of the target arm against (a) UCB algorithm, (b) Thompson sampling algorithm, and (c) ϵ-greedy algorithm.

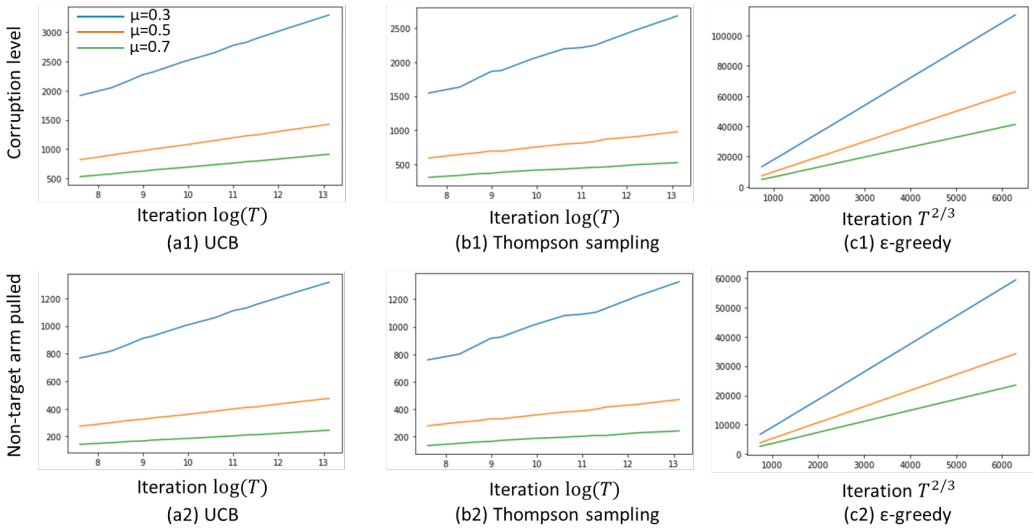


Figure A.2: The modified attack which knows the mean reward of the target arm against (a1),(a2) UCB algorithm, (b1),(b2) Thompson sampling algorithm, and (c1),(c2) ϵ-greedy algorithm.

APPENDIX B

MISSING PROOFS AND ADDITIONAL EXPERIMENTS FROM CHAPTER 5

B.1 Characterizing the optimal pacing strategy and budget allocation in expectation

Recalling that the optimal strategy on the realized values and prices is obtained by the hindsight strategy H (definition 5.2.3). The Lagrangian dual of the optimization problem in definition 5.2.3 is given by:

$$\psi(\mu) = \left[\sum_{t=1}^{T} \left(v^t - (1+\mu)p^t\right)^+\right] + \mu B \qquad \text{(L(KP))} \tag{B.1}$$

Where we define $(z)^+$ to be $\max\{z, 0\}$. The dual is obtained from the Lagrangian by setting $x^t = 1$ for all t such that $v^t - (1+\mu)p^t \geq 0$, i.e. winning all impressions with value greater than $(1+\mu)$ times the price which can be done by bidding $b^t = v^t/(1+\mu)$.

By weak duality, we have

$$\pi^H(\boldsymbol{v}, \boldsymbol{p}) \leq \inf_{\mu \geq 0} \psi(\mu) \tag{B.2}$$

Since $\boldsymbol{v}$ and $\boldsymbol{p}$ are being sampled from the fixed distribution defined by $\vec{Q}$, taking expectation over equation B.2 and using Jensen's inequality, we get

$$\mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\pi^H(\boldsymbol{v}, \boldsymbol{p})\right] \leq \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\inf_{\mu \geq 0} \psi(\mu)\right] \leq \inf_{\mu \geq 0} \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}[\psi(\mu)] \tag{B.3}$$

Let $\Psi(\mu) = \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}[\psi(\mu)]$ and $\mu^\star$ be the minimizer of $\Psi(\mu)$. Assuming $\Psi(\mu)$ to be differentiable, using Karush-Kuhn-Tucker conditions, we have $\mu^\star \geq 0$, $\Psi'(\mu^\star) \geq 0$, and $\mu^\star \Psi'(\mu^\star) = 0$. If $\mu^\star = 0$, it implies that we are effectively not constrained by the budget and truthful bidding achieves the optimal utility in expectation as it wins all items with positive utility.

The gradient of $\Psi(\mu)$ can be written as

$$\Psi'(\mu) = B - G(\mu)$$

where

$$G(\mu) = \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}}\left[\sum_{t=1}^{T} \mathbb{1}\left\{v^t \geq (1+\mu)p^t\right\} p^t\right]$$

We call $G(\mu)$ the overall spend function. By definition, $G(\mu)$ is the expected expenditure over all the T rounds when buying all items such that $v^t \geq (1+\mu)p^t$, obtained by bidding $b^t = v^t/(1+\mu)$. The KKT complementary slackness condition implies that if $\mu^\star > 0$, then $\Psi'(\mu^\star) = 0$ i.e.

$$G(\mu^\star) = B \tag{B.4}$$

This implies that the strategy with a fixed pacing multiplier that bids $b^t = v^t/(1+\mu^\star)$ achieves better expected utility than the expected utility of the hindsight strategy H. If truthful bidding is not optimal (i.e $\mu^\star > 0$), then the expected expenditure of this strategy is B. Not that the expenditure guarantee for the optimal fixed shading strategy is only satisfied in expectation, i.e. it spends budget B in expectation.

For the rest of the theoretical claims, we restrict ourselves to the case that $\mu^\star > 0$. The case $\mu^\star = 0$ implies that the budget constraint is not binding, so truthful bidding is the optimal strategy. Our algorithm will naturally adapt to this setting as well.

Let $\overline{G}(\mu) = \frac{G(\mu)}{T}$ be the average spend in each round (over the whole campaign) if we buy all items such that $v^t \geq (1+\mu)p^t$. Equation B.4 implies that for the optimal dual variable $\mu^\star > 0$,

$$\overline{G}(\mu^\star) = \frac{B}{T}. \tag{B.5}$$

Using Definition 5.2.7, $\overline{G_e}(\mu) = \mathbf{E}_{(v,p)\sim Q_e}[\mathbb{1}\left\{v \geq (1+\mu)p\right\} p]$. $\overline{G_e}(\mu)$ is the expected spend per round in an episode e if the dual variable is μ, corresponding to the strategy which bids by multiplicatively shading the value v^t by a factor of $\frac{1}{1+\mu}$ and ends up buying all the

impressions in the episode with value per unit spent at least $(1+\mu)$. In our framework, the spend function can be decomposed across the episodes by introducing episodic spend functions. We show this decomposition below:

$$
\begin{aligned}
G(\mu) &= \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}\sim\vec{Q}}\left[\sum_{t=1}^{T} \mathbb{1}\left\{v^t \geq (1+\mu)p^t\right\} p^t\right] \\
&= \mathbf{E}_{\boldsymbol{v},\boldsymbol{p}\sim\vec{Q}}\left[\sum_{e=1}^{E} \sum_{t=(e-1)\tau+1}^{e\tau} \mathbb{1}\left\{v^t \geq (1+\mu)p^t\right\} p^t\right] \\
&= \sum_{e=1}^{E} \sum_{t=(e-1)\tau+1}^{e\tau} \mathbf{E}_{(v^t,p^t)\sim Q_e}\left[\mathbb{1}\left\{v^t \geq (1+\mu)p^t\right\} p^t\right] \\
&= \tau \sum_{e=1}^{E} \mathbf{E}_{(v,p)\sim Q_e}[\mathbb{1}\left\{v \geq (1+\mu)p\right\} p] \\
&= \tau \sum_{e=1}^{E} \overline{G_e}(\mu) \\
\implies \frac{G(\mu)}{T} &= \frac{1}{E}\sum_{e=1}^{E} \overline{G_e}(\mu) \\
\implies \overline{G}(\mu) &= \frac{1}{E}\sum_{e=1}^{E} \overline{G_e}(\mu)
\end{aligned}
\tag{B.6}
$$

where $\overline{G_e}(\mu) = \mathbf{E}_{(v,p)\sim Q_e}[\mathbb{1}\left\{v \geq (1+\mu)p\right\} p]$ is the episodic spend function. This definition helps us to define the optimal budget allocation as $B_e = \tau\overline{G_e}(\mu^\star) = \tau\rho_e$ where ρ_e is the *optimal spend rate* for episode e given by $\rho_e = \overline{G_e}(\mu^\star)$. Note that if $\mu^\star > 0$, using Equation 5.5, we have

$$
\tau \sum_{e=1}^{E} \rho_e = \tau \sum_{e=1}^{E} \overline{G_e}(\mu^\star) = G(\mu^\star) = B \tag{B.7}
$$

B.2 Detailed Algorithms

B.2.1 EpisodicAdaptivePacing: Adaptive pacing using a spend plan

We present ApproxSpendRate (Algorithm 6) which uses historical data to compute approximately optimal spend rates $(\widehat{\rho}_1, \ldots, \widehat{\rho}_E)$ from samples.

Algorithm 16: EpisodicAdaptivePacing: Adaptive pacing using a spend plan.

Input: Budget B, rounds T, episodes E, spend plan $(\rho'_1, \ldots, \rho'_E)$, step size η, max shading param $\bar{\mu}$

$\mu_i \leftarrow [0, \bar{\mu}]$ // Initialize shading multiplier

$\textsc{Budget}_1 \leftarrow B$ // Overall remaining budget left for campaign

$\tau \leftarrow \frac{T}{E}$ // Impressions in each episode

$\widehat{B}_1 \leftarrow \rho'_1 \cdot \tau$ // Remaining budget for episode 1

for $t = 1, \ldots, T$ **do**

  $e \leftarrow \lceil t/E \rceil$ // Current episode

  Observe value v^t

  Post bid $b^t \leftarrow \min\left\{\frac{v^t}{1+\mu^t}, \widehat{B_e}, \textsc{Budget}^t\right\}$

  Observe expenditure z^t

  $\mu^{t+1} \leftarrow \textsc{Proj}_{[0,\bar{\mu}]}[\mu^t - \eta(\rho'_e - z^t)]$ // Update shading parameter

  $\widehat{B}_e \leftarrow \widehat{B}_e - z^t$ // Update remaining budget

  $\textsc{Budget}^{t+1} \leftarrow \textsc{Budget}^t - z^t$

  if $t \pmod E = 0$ **then**

    $\widehat{B}_{e+1} \leftarrow \rho'_{e+1} \cdot \tau + \widehat{B}_e$ // Carry over left-over budget

  end

end

For each episode e, the subroutine ApproxSpendSP (Algorithm 7) estimates the episodic spend function $\overline{G_e}(\mu)$ as a function of μ using the historic samples $\vec{V}$ and $\vec{P}$. For fixed prices, we use a simpler episodic spend prediction function estimate ApproxSpendFP (Algorithm 8). Both functions try to estimate $\overline{G_e}(\mu)$ and return an empirical approximate of $\overline{G_e}(\mu)$ we denote as $\widehat{G_e}(\mu)$.

Then using the structure of overall average spend function $\overline{G}(\mu)$, (Equation B.6), we can construct an approximation of the overall average spend function $\widehat{G}(\mu)$ as $\frac{\sum_{e=1}^{E} \widehat{G_e}(\mu)}{E}$. Based on our discussion about the optimal structure of the problem, for optimal dual variable $\mu^\star$, we know that $\overline{G}(\mu^\star) = \frac{B}{T}$ (Equation 5.5). Using our empirical estimate $\widehat{G}(\mu)$, we compute $\widehat{\mu}$, an empirical estimate of $\mu^\star$. We can compose our approximations to form $\widehat{\rho}_e = \widehat{G_e}(\widehat{\mu})$, an approximation to $\overline{G_e}(\mu^\star) = \rho_e$.

Algorithm 17: ApproxSpendRate: Approximate optimal spend rates

Input: Budget B, Total rounds T, Number of episodes E, Episodic sampling oracles F_e for values and D_e for prices, Per episode sampling budget n, Kernel K, scalar s

Goal: Estimate optimal spend rates $(\rho_1, \ldots, \rho_E)$

```
if in the constant-price setting then
    for e = 1, ..., E do
        Samples n values V⃗ = (V_1, V_2, ..., V_n) ~ F_e
        Set price p ~ D_e
        Ĝ_e(μ) = ApproxSpendFP(n, V⃗, p)          // Estimate episodic spend function
    end
end
else
    for e = 1, ..., E do
        Samples n values V⃗ = (V_1, V_2, ..., V_n) ~ F_e
        Samples n prices P⃗ = (P_1, P_2, ..., P_n) ~ D_e
        Ĝ(μ) = ApproxSpendSP(n, V⃗, P⃗, K, s)      // Estimate episodic spend function
    end
end
Ḡ_e(μ) = (Σ_{e=1}^{E} Ĝ_e(μ)) / E                // Construct overall average spend function
μ̂ = min μ s.t. Ḡ_e(μ) ≤ B/T                      // Estimating the optimal dual variable
for e = 1, ..., E do
    ρ̂_e = Ĝ_e(μ̂)                                // Expected spend rate in episode for the estimated dual variable
end
return (ρ̂_1, ..., ρ̂_E)
```

Table B.1: Descriptions of synthetic datasets used for experiments.

Data set	Value distributions F_e	Price distributions D_e
uniform_v_fix_p	Uniform dist over $[l_e, r_e]$	Fixed price p
normal_v_fix_p	$\mathcal{N}(\mu_{v,e}, \sigma^2_{v,e})$	Fixed price p
lognorm_v_fix_p	Lognormal$(\mu_{v,e}, \sigma^2_{v,e})$	Fixed price p
uniform_v_normal_p	Uniform dist over $[l_e, r_e]$	$\mathcal{N}(\mu_{p,e}, \sigma^2_{p,e})$
normal_v_normal_p	$\mathcal{N}(\mu_{v,e}, \sigma^2_{v,e})$	$\mathcal{N}(\mu_{p,e}, \sigma^2_{p,e})$
lognorn_v_maxlognorm_p	Lognormal$(\mu_{v,e}, \sigma^2_{v,e})$	$\max_{k\in[K]}$ Lognormal$(\mu_{k,e}, \sigma^2_{k,e})$

B.3 Experiment on Synthetic Data

We now consider synthetic datasets that meet the definition of admissable distributions from Definition 5.4.1.

B.3.1 Datasets

We create synthetic datasets to test the performance of the algorithms under consideration. For the values, we consider three distributions: uniform, normal, and lognormal. For the prices, we consider three settings: fixed prices (our analysis focuses on this setting before generalizing), normally distributed prices, and the max of multiple draws from a lognormal distribution.[1] We combine these into 6 synthetic datasets, see Table B.1. We divide the time horizon into 10 episodes, i.e. $E = 10$ with differing parameters of distributions for each episode and use $T = 1000$.

B.3.2 Results

The values and prices were generated in the same way as above. For each dataset, we run simulations where we a budget for the campaign is drawn uniformly from $[0, \bar{C}]$, where $\bar{C}$ is the expenditure of the campaign that bids truthfully in each auction. Then we run

[1]Prior work, e.g. [147], suggest that bids in ad auctions typically follow a lognormal distribution. The combination of lognormal values with max-of-lognormal-draws as prices is a realistic simulation of auction environment which is captured in the lognorn_v_maxlognorm_p dataset.

all the pacing algorithms for this dataset sample and budget level. We repeat this process 150 times to get 150 data points per dataset for each algorithm. We plot the ratio of the optimal utility that the pacing system is able to obtain as a function of the budget level in Figure B.2. Our algorithm outperforms both benchmarks almost everywhere. The only time where the "Truthful" benchmark performs better are in situations where the advertiser has enough budget to buy (almost) all impressions. There is one area where "Fixed spend (BG19)" outperforms our algorithm. It happens for the "normal_v_normal_p" dataset when $B \geq 0.8 \cdot \bar{C}$; we do not have an explanation why this particular range performs poorly.

Efficiency of Offline Training To understand how the performance of our end-to-end pacing is dependent on the numbers of samples available in the spend plan estimation phase, we plot the ratio of the optimal utility that the pacing system is able to obtain as a function of the number of training samples in Figure B.1. We consider 4 different budget levels: let $\bar{C}$ be the expenditure of the campaign that bids their value in each auction, we consider budgets $x\bar{C}$ for $x \in \{0.25, 0.5, 0.75, 1.0\}$.

Figure B.1 shows the effect of varying the training set size, where a sample in the training set corresponds to a value and price draw from each episode. Two things are clear from the results: 1) with increasing samples, the performance improves quickly, and 2) the budget level is important for the overall performance as the optimal budget allocation problem gets harder for smaller budgets. Finally, it does appear that the performance hits a plateau. This is likely due to the online part of the algorithm which does not scale with increased offline learning sample size.

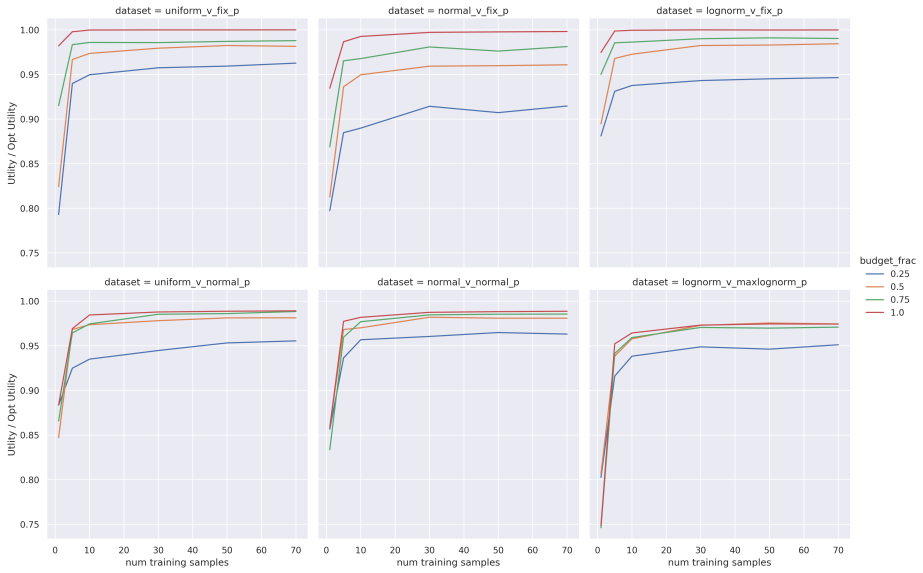


Figure B.1: Performance of the end-to-end pacing system as a function of the size of training data. We plot the ratio of the optimal utility that our pacing system is able to obtain as a function of the number of training samples in the rate rate estimation phase. The budget is represented as budget_frac, i.e. $B = \text{budget\_frac}*\bar{C}$. We can see that 1) with increasing samples, the performance improves quickly 2) the budget level is important for the overall performance.

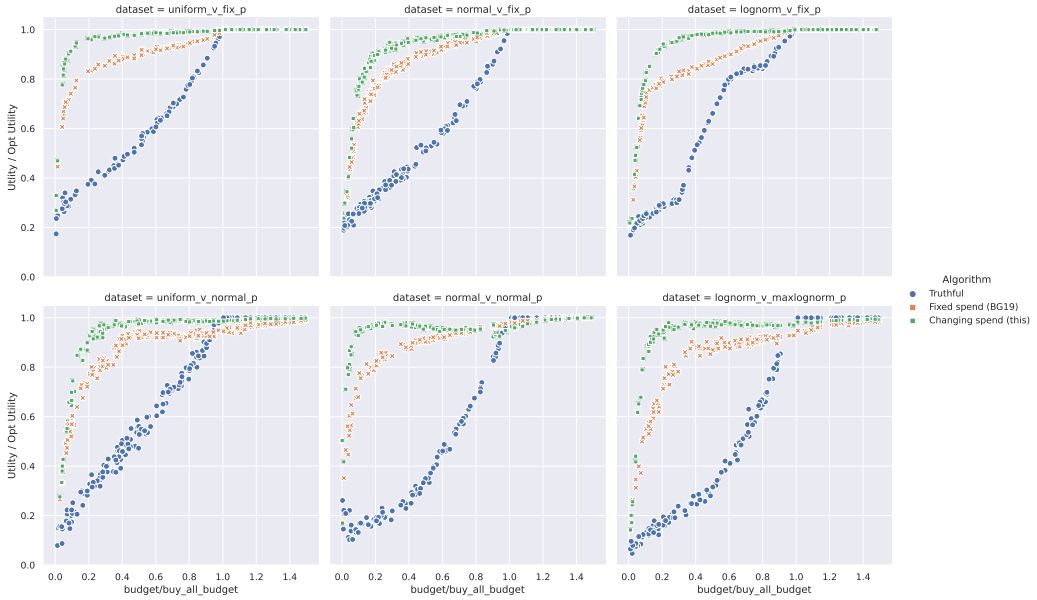


Figure B.2: Comparing performance of our algorithm labeled as *Changing spend (this)*, fixed spend rate [81] labeled as *Fixed spend (BG19)* and no pacing labeled as *Truthful* on synthetic datasets. See Table B.1 for details on datasets. Each datapoint in the scatter plot refers to one experiment where we plot the fraction of the optimal utility obtained by the pacing strategy as a function of the budget *buy_all_budget* represents $\bar{C}$. In each of these cases, our method achieves a higher fraction of optimal utility than either no pacing (truthful bidding) or fixed spend rate pacing strategies ([81]) over nearly all ratios of the budget relative to the cost of all impressions.

REFERENCES

[1] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz, "Minimizing regret with label efficient prediction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 2152–2162, 2005.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002. eprint: https://doi.org/10.1137/S0097539701398375.

[3] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: A meta-algorithm and applications," *Theory of computing*, vol. 8, no. 1, pp. 121–164, 2012.

[4] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[5] J. D. Abernethy, R. Cummings, B. Kumar, S. Taggart, and J. H. Morgenstern, "Learning auctions with robust incentive guarantees," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 587–11 597.

[6] Y. Xu, B. Kumar, and J. D. Abernethy, "Observation-free attacks on stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[7] J. Abernethy, B. Kumar, T. Lykouris, and Y. Xu, "Bridging truthfulness and corruption-robustness in multi-armed bandit mechanisms," *Incentives in Machine Learning Workshop, ICML 2020*,

[8] B. Kumar, J. Morgenstern, and O. Schrijvers, "Optimal spend rate estimation and pacing for ad campaigns with budgets," *arXiv preprint arXiv:2202.05881*, 2022.

[9] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*, Springer, 1995, pp. 23–37.

[10] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Information and computation*, vol. 108, no. 2, pp. 212–261, 1994.

[11] S. Hanneke, "A bound on the label complexity of agnostic active learning," in *Proceedings of the 24th international conference on Machine learning*, ACM, 2007, pp. 353–360.

[12] C. Zhang, "Efficient active learning of sparse halfspaces," arXiv preprint, 2018. arXiv: 1805.02350.

[13] S. Hanneke and L. Yang, "Surrogate losses in passive and active learning," arXiv preprint, 2012. arXiv: 1207.3772.

[14] S. Hanneke, "Rates of convergence in active learning," *The Annals of Statistics*, pp. 333–361, 2011.

[15] S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis, "Active learning using arbitrary binary valued queries," *Machine Learning*, vol. 11, no. 1, pp. 23–35, 1993.

[16] V. Koltchinskii, "Local rademacher complexities and oracle inequalities in risk minimization," *The Annals of Statistics*, vol. 34, no. 6, pp. 2593–2656, 2006.

[17] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[18] S. Dasgupta, D. J. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," *In Advances in neural information processing systems*, pp. 353–360, 2008.

[19] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 65–72.

[20] B. Kumar, J. D. Abernethy, and V. Saligrama, "Activehedge: Hedge meets active learning," in *International Conference on Machine Learning*, PMLR, 2022, pp. 11 694–11 709.

[21] L. Wang, Y. Bai, W. Sun, and T. Joachims, "Fairness of exposure in stochastic bandits," in *International Conference on Machine Learning (ICML)*, 2021.

[22] B. Kumar, M. Kleindessner, J. Abernethy, and M. Kearns, "Group fairness of exposure in multi-arm bandits," *Under Preparation*, 2022.

[23] E. Elkind, "Designing and learning optimal finite support auctions," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07, New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 736–745, ISBN: 978-0-898716-24-5.

[24] Y. A. Gonczarowski and N. Nisan, "Efficient empirical revenue maximization in single-parameter auction environments," in *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, ser. STOC '17, 2017, pp. 856–868.

[25] R. Cole and T. Roughgarden, "The sample complexity of revenue maximization," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, ACM, 2014, pp. 243–252.

[26] M.-F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour, "Mechanism design via machine learning," in *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, IEEE, 2005, pp. 605–614.

[27] P. Dhangwatnotai, T. Roughgarden, and Q. Yan, "Revenue maximization with a single sample," *Games and Economic Behavior*, vol. 91, pp. 318–333, 2015.

[28] J. Morgenstern and T. Roughgarden, "Learning simple auctions," in *Conference on Learning Theory*, 2016, pp. 1298–1318.

[29] M.-F. Balcan, T. Sandholm, and E. Vitercik, "Sample complexity of automated mechanism design," in *Advances in Neural Information Processing Systems*, 2016, pp. 2083–2091.

[30] J. H. Morgenstern and T. Roughgarden, "On the pseudo-dimension of nearly optimal auctions," in *Advances in Neural Information Processing Systems*, 2015, pp. 136–144.

[31] J. Hartline and S. Taggart, *Non-revelation mechanism design*, arXiV preprint 1608.01875, 2016.

[32] N. R. Devanur, Z. Huang, and C.-A. Psomas, "The sample complexity of auctions with side information," in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, ACM, 2016, pp. 426–439.

[33] R. Cummings, "Differential privacy as a tool for truthfulness in games," *XRDS*, vol. 24, no. 1, pp. 34–37, Sep. 2017.

[34] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, IEEE, 2007, pp. 94–103.

[35] K. Nissim, C. Orlandi, and R. Smorodinsky, "Privacy-aware mechanism design," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12, ACM, 2012, pp. 774–789.

[36] M. Kearns, M. Pai, A. Roth, and J. Ullman, "Mechanism design in large games: Incentives and privacy," in *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ser. ITCS '14, ACM, 2014, pp. 403–410.

[37] R. Cummings, M. Kearns, A. Roth, and Z. S. Wu, "Privacy and truthful equilibrium selection for aggregative games," in *Proceedings of the 11th International Conference on Web and Internet Economics*, ser. WINE '15, 2015, pp. 286–299.

[38] R. Cummings, S. Ioannidis, and K. Ligett, "Truthful linear regression," in *Proceedings of The 28th Conference on Learning Theory*, ser. COLT '15, 2015, pp. 448–483.

[39] S. Kannan, J. Morgenstern, R. Rogers, and A. Roth, "Private pareto optimal exchange," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, ser. EC '15, 2015, pp. 261–278.

[40] D. Xiao, "Is privacy compatible with truthfulness?" In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, ACM, 2013, pp. 67–86.

[41] Y. Chen, S. Chong, I. A. Kash, T. Moran, and S. Vadhan, "Truthful mechanisms for agents that value privacy," *ACM Transactions on Economics and Computation (TEAC)*, vol. 4, no. 3, p. 13, 2016.

[42] J. Liu, Z. Huang, and X. Wang, "Learning optimal reserve price against non-myopic bidders," in *Advances in Neural Information Processing Systems*, 2018, pp. 2038–2048.

[43] K. Amin, A. Rostamizadeh, and U. Syed, "Learning prices for repeated auctions with strategic buyers," in *Advances in Neural Information Processing Systems*, 2013, pp. 1169–1177.

[44] J. Hartline, "Mechanism design and approximation," *Book draft. October*, vol. 122, 2013.

[45] D. Bergemann and J. Valimaki, "The dynamic pivot mechanism," *Econometrica*, vol. 78, no. 2, pp. 771–789, 2010.

[46] S. Kakade, I. Lobel, and H. Nazerzadeh, "Optimal dynamic mechanism design and the virtual-pivot mechanism," *Operations Research*, vol. 64, no. 4, pp. 837–854, 2013.

[47] A. Pavan, I. Segal, and J. Toikka, "Dynamic mechanism design: A Myersonian approach," *Econometrica*, vol. 82, no. 2, pp. 601–653, 2014.

[48] Y. Kanoria and H. Nazerzadeh, "Dynamic reserve prices for repeated auctions: Learning from bids," *arXiv preprint arXiv:2002.07331*, 2020.

[49] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, Springer, 2006, pp. 265–284.

[50] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, p. 26, 2011.

[51] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*, ACM, 2010, pp. 715–724.

[52] P. Massart, "The tight constant in the dvoretzky-kiefer-wolfowitz inequality," *The annals of Probability*, pp. 1269–1283, 1990.

[53] D. Bouneffouf, A. Bouzeghoub, and A. L. Gançarski, "A contextual-bandit algorithm for mobile context-aware recommender system," in *International conference on neural information processing*, Springer, 2012, pp. 324–331.

[54] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, "Scene: A scalable two-stage personalized news recommendation system," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 125–134.

[55] J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla, "Efficient thompson sampling for online matrix-factorization recommendation," in *Advances in neural information processing systems*, 2015, pp. 1297–1305.

[56] L. Li, W. Chu, J. Langford, and X. Wang, "Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 297–306.

[57] S. S. Villar, J. Wason, and J. Bowden, "Response-adaptive randomization for multi-arm clinical trials using the forward looking gittins index rule," *Biometrics*, vol. 71, no. 4, pp. 969–978, 2015.

[58] E. M. Schwartz, E. T. Bradlow, and P. S. Fader, "Customer acquisition via display advertising using multi-armed bandit experiments," *Marketing Science*, vol. 36, no. 4, pp. 500–522, 2017.

[59] H. Haddadi, "Fighting online click-fraud using bluff ads," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 21–25, 2010.

[60] K. C. Wilbur and Y. Zhu, "Click fraud," *Marketing Science*, vol. 28, no. 2, pp. 293–308, 2009.

[61] N. Kshetri, "The economics of click fraud," *IEEE Security & Privacy*, vol. 8, no. 3, pp. 45–53, 2010.

[62] T. Lappas, "Fake reviews: The malicious perspective," in *International Conference on Application of Natural Language to Information Systems*, Springer, 2012, pp. 23–34.

[63] T. Lappas, G. Sabnis, and G. Valkanas, "The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry," *Information Systems Research*, vol. 27, no. 4, pp. 940–961, 2016.

[64] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[65] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[66] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[67] H. Dai *et al.*, "Adversarial attack on graph structured data," *arXiv preprint arXiv:1806.02371*, 2018.

[68] X. Liu, S. Si, X. Zhu, Y. Li, and C.-J. Hsieh, "A unified framework for data poisoning attack to graph-based semi-supervised learning," *arXiv preprint arXiv:1910.14147*, 2019.

[69] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[70] K.-S. Jun, L. Li, Y. Ma, and J. Zhu, "Adversarial attacks on stochastic bandits," in *Advances in Neural Information Processing Systems*, 2018, pp. 3640–3649.

[71] F. Liu and N. Shroff, "Data poisoning attacks on stochastic bandits," *arXiv preprint arXiv:1905.06494*, 2019.

[72] Y. Ma, K.-S. Jun, L. Li, and X. Zhu, "Data poisoning attacks in contextual bandits," in *International Conference on Decision and Game Theory for Security*, Springer, 2018, pp. 186–204.

[73] E. Garcelon *et al.*, "Adversarial attacks on linear contextual bandits," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[74] T. Lykouris, V. Mirrokni, and R. Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 114–122.

[75] A. Gupta, T. Koren, and K. Talwar, "Better algorithms for stochastic bandits with adversarial corruptions," *arXiv preprint arXiv:1902.08647*, 2019.

[76] S. Kapoor, K. K. Patel, and P. Kar, "Corruption-tolerant bandit learning," *Machine Learning*, vol. 108, no. 4, pp. 687–715, 2019.

[77] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on learning theory*, 2012, pp. 39–1.

[78] N. R. Devanur and S. M. Kakade, "The price of truthfulness for pay-per-click auctions," in *Proceedings of the 10th ACM conference on Electronic commerce*, 2009, pp. 99–106.

[79] M. Babaioff, Y. Sharma, and A. Slivkins, "Characterizing truthful multi-armed bandit mechanisms," *SIAM J. Comput.*, vol. 43, no. 1, pp. 194–230, 2014.

[80] M. Babaioff, Y. Sharma, and A. Slivkins, "Characterizing truthful multi-armed bandit mechanisms," *SIAM J. Comput.*, vol. 43, no. 1, pp. 194–230, 2014.

[81] S. R. Balseiro and Y. Gur, "Learning in repeated auctions with budgets: Regret minimization and equilibrium," *Management Science*, vol. 65, no. 9, pp. 3952–3968, 2019.

[82] Statista, *Online advertising revenue in the united states from 2000 to 2020*, https://www.statista.com/statistics/183816/us-online-advertising-revenue-since-2000/, 2020.

[83] J. Liu and S. Hill, "Moment marketing: Measuring dynamics in cross-channel ad effectiveness," *Available at SSRN 3670024*, 2020.

[84] D. Agarwal, S. Ghosh, K. Wei, and S. You, "Budget pacing for targeted online advertisements at linkedin," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1613–1619.

[85] X. Ma *et al.*, "Large-scale user visits understanding and forecasting with deep spatial-temporal tensor factorization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2403–2411.

[86] D. Agarwal, D. Chen, L.-j. Lin, J. Shanmugasundaram, and E. Vee, "Forecasting high-dimensional data," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 1003–1012.

[87] K.-C. Lee, A. Jalali, and A. Dasdan, "Real time bid optimization with smooth budget delivery in online advertising," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, 2013, pp. 1–9.

[88] S. R. Balseiro and Y. Gur, "Learning in repeated auctions with budgets: Regret minimization and equilibrium," in *Proceedings of the 2017 ACM Conference on Economics and Computation*, ser. EC '17, Cambridge, Massachusetts, USA: ACM, 2017, pp. 609–609, ISBN: 978-1-4503-4527-9.

[89] P. Rusmevichientong and D. P. Williamson, "An adaptive algorithm for selecting profitable keywords for search-based advertising services," in *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*, 2006, pp. 260–269.

[90] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein, "Budget optimization in search-based advertising auctions," in *Proceedings of the 8th ACM conference on Electronic commerce*, 2007.

[91] K. Hosanagar and V. Cherepanov, "Optimal bidding in stochastic budget constrained slot auctions," in *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*, 2008, p. 20.

[92] V. Conitzer, C. Kroer, E. Sodomka, and N. E. Stier-Moses, "Multiplicative pacing equilibria in auction markets," in *Conference on Web and Internet Economics (WINE'18)*, Oxford, UK, 2018.

[93] S. Balseiro, H. Lu, and V. Mirrokni, "Dual mirror descent for online allocation problems," in *International Conference on Machine Learning*, PMLR, 2020, pp. 613–628.

[94] V. Conitzer *et al.*, "Pacing equilibrium in first-price auction markets," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, ser. EC '19, Phoenix, AZ, USA: ACM, 2019, pp. 587–587, ISBN: 978-1-4503-6792-9.

[95] Y. Gao, C. Kroer, and A. Peysakhovich, *Online market equilibrium with application to fair division*, 2021. arXiv: 2103.12936 [cs.GT].

[96] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and generalized online matching," *Journal of the ACM (JACM)*, vol. 54, no. 5, 2007.

[97] Z. Abrams, S. S. Keerthi, O. Mendelevitch, and J. A. Tomlin, "Ad delivery with budgeted advertisers: A comprehensive lp approach.," *Journal of Electronic Commerce Research*, vol. 9, no. 1, 2008.

[98] Y. Azar, B. Birnbaum, A. R. Karlin, and C. T. Nguyen, "On revenue maximization in second-price ad auctions," in *Algorithms - ESA 2009*, A. Fiat and P. Sanders, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 155–166, ISBN: 978-3-642-04128-0.

[99] A. Goel, M. Mahdian, H. Nazerzadeh, and A. Saberi, "Advertisement allocation for generalized second-pricing schemes," *Oper. Res. Lett.*, vol. 38, no. 6, pp. 571–576, Nov. 2010.

[100] C. Karande, A. Mehta, and R. Srikant, "Optimizing budget constrained spend in search advertising," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM '13, Rome, Italy: ACM, 2013, pp. 697–706, ISBN: 978-1-4503-1869-3.

[101] K. Amin, M. Kearns, P. Key, and A. Schwaighofer, "Budget optimization for sponsored search: Censored learning in mdps," *arXiv preprint arXiv:1210.4847*, 2012.

[102] L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings, "Knapsack based optimal policies for budget–limited multi–armed bandits," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[103] A. Flajolet and P. Jaillet, "Real-time bidding with side information," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., 2017, pp. 5168–5178.

[104] A. Nuara, F. Trovò, N. Gatti, and M. Restelli, *Online joint bid/daily budget optimization of internet advertising campaigns*, 2020. arXiv: 2003.01452 [cs.LG].

[105] V. Avadhanula, R. Colini-Baldeschi, S. Leonardi, K. Abinav Sankararaman, and O. Schrijvers, "Stochastic bandits for multi-platform budget optimization in online advertising," in *The World Wide Web Conference*, 2020.

[106] M. Cary *et al.*, "Greedy bidding strategies for keyword auctions.," in *EC*, J. K. MacKie-Mason, D. C. Parkes, and P. Resnick, Eds., ACM, 2007, pp. 262–271, ISBN: 978-1-59593-653-0.

[107] C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, and M. Mahdian, "Dynamics of bid optimization in online advertisement auctions," in *Proceedings of the 16th international conference on World Wide Web*, 2007.

[108] S. Balseiro, A. Kim, M. Mahdian, and V. Mirrokni, "Budget management strategies in repeated auctions," in *Proceedings of the 26th International World Wide Web Conference, Perth, Australia*, 2017.

[109] M. Babaioff, R. Cole, J. Hartline, N. Immorlica, and B. Lucier, "Non-quasi-linear agents in quasi-linear mechanisms," *arXiv preprint arXiv:2012.02893*, 2020.

[110] X. Chen, C. Kroer, and R. Kumar, *The complexity of pacing for second-price auctions*, 2021. arXiv: 2103.13969 [cs.GT].

[111] S. Balseiro, A. Kim, M. Mahdian, and V. Mirrokni, "Budget-constrained incentive compatibility for stationary mechanisms," in *Proceedings of the 21st ACM Conference on Economics and Computation*, 2020, pp. 607–608.

[112] S. M. Kakade, A. T. Kalai, and K. Ligett, "Playing games with approximation algorithms," *SIAM Journal on Computing*, vol. 39, no. 3, pp. 1088–1106, 2009.

[113] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

[114] R. A. Davis, K.-S. Lii, and D. N. Politis, "Remarks on some nonparametric estimates of a density function," in *Selected Works of Murray Rosenblatt*, Springer, 2011, pp. 95–100.

[115] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[116] H. Jiang, "Uniform convergence rates for kernel density estimation," in *International Conference on Machine Learning*, PMLR, 2017, pp. 1694–1703.

[117] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[118] B. Settles, "From theories to queries: Active learning in practice," in *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, 2011, pp. 1–18.

[119] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 79.

[120] M. Wang and X.-S. Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 2, pp. 1–21, 2011.

[121] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[122] X. Li and Y. Guo, "Adaptive active learning for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 859–866.

[123] S. Hanneke, "Theory of disagreement-based active learning," Foundations and Trends® in Machine Learning, vol. 7, no. 2-3, pp. 131–309, 2014.

[124] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[125] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[126] A. Beygelzimer, S. Dasgupta, and J. Langford, "Importance weighted active learning," *ArXiv*, vol. abs/0812.4952, 2008.

[127] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-case analysis of selective sampling for linear classification," *Journal of Machine Learning Research*, vol. 7, pp. 1205–1230, Jul. 2006.

[128] N. Cesa-Bianchi, C. Gentile, and F. Orabona, "Robust bounds for classification via selective sampling," in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 121–128.

[129] O. Dekel, C. Gentile, and K. Sridharan, "Selective sampling and active learning from single and multiple teachers," *Journal of Machine Learning Research*, vol. 13, pp. 2655–2697, Sep. 2012.

[130] A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang, "Agnostic active learning without constraints," *Advances in Neural Information Processing Systems*, pp. 199–207, 2010.

[131] S. Hanneke, *Adaptive rates of convergence in active learning*. In *COLT*. Citeseer, 2009.

[132] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," in International Conference on Computational Learning Theory, 2005, pp. 249–263.

[133] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in International Conference on Computational Learning Theory, 2007, pp. 35–50.

[134] M.-F. Balcan and P. Long, "Active and passive learning of linear separators under log-concave distributions," in Conference on Learning Theory, 2013, pp. 288–316.

[135] P. Awasthi, M. F. Balcan, and P. M. Long, "The power of localization for efficiently learning linear separators with noise," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, ACM, 2014, pp. 449–458.

[136] P. Awasthi, M.-F. Balcan, N. Haghtalab, and R. Urner, "Efficient learning of linear separators under bounded noise," in Conference on Learning Theory, 2015, pp. 167–190.

[137] C. Cortes, G. DeSalvo, C. Gentile, M. Mohri, and N. Zhang, "Region-based active learning," in The 22nd International Conference on Artificial Intelligence and Statistics, 2019, pp. 2801–2809.

[138] L. Yang, "Active learning with a drifting distribution," *In Advances in Neural Information Processing Systems*, pp. 2079–2087, 2011.

[139] P. Zhao, S. Hoi, and J. Zhuang, "Active learning with expert advice," *arXiv preprint arXiv:1309.6875*, 2013.

[140] S. Hao, P. Hu, P. Zhao, S. C. Hoi, and C. Miao, "Online active learning with expert advice," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 5, pp. 1–22, 2018.

[141] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[142] S. Hanneke and L. Yang, "Minimax analysis of active learning," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3487–3602, 2015.

[143] D. Sculley, "Online active learning methods for fast label-efficient spam filtering.," in *CEAS*, vol. 7, 2007, p. 143.

[144] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[145] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[146] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[147] M. Ostrovsky and M. Schwarz, "Reserve prices in internet advertising auctions: A field experiment," in *Proceedings of the 12th ACM conference on Electronic commerce*, 2011, pp. 59–60.