Tel Aviv University

The Raymond and Beverly Sackler Faculty of Exact Sciences

The Blavatnik School of Computer Science

Topics in Machine Learning, Reinforcement Learning and Societal Challenges

Thesis submitted for the degree

"Doctor of Philosophy"

Lee Cohen

under the supervision of **Professor Yishay Mansour**

Submitted to the Senate of Tel Aviv University

September 2022

Acknowledgements

First and foremost, I would like to express deep gratitude to my thesis advisor, Yishay Mansour, for being the best Ph.D. (and MS.c.) mentor I could have asked for. I owe much of my research knowledge and success to you. Throughout the way, you have continually inspired me with your profound insights, quick thinking, ability to identify the right research questions, and ways to pursue solutions. I'm still amazed by how naturally you have managed to balance between providing me guidance and yet encouraging me to be an independent researcher. Thank you for always being available when I needed you, providing valuable feedback, and offering helpful suggestions for improvement, all the while making me enjoy the journey. I feel incredibly privileged to have you as my advisor, and truly hope to become half as great of a mentor as you have been to me!

During my Ph.D., I have been fortunate enough to have mentors who generously shared their wisdom, and insights, and acted as role models. They have supported and encouraged me to push beyond my limits to achieve my full potential. I am truly grateful to them. Omer Ben-Porat provided me with insightful feedback and advice on conducting research. Zack Lipton taught me how to translate real-world problems into research questions, supported me from the very first research project of my Ph.D., and shared invaluable techniques on paper writing. Michal Moshkovitz

encouraged me to pursue my goals and showed me the power of optimism in research. Additionally, I extend my gratitude to Tomer Koren and Guy Rothblum for their supportive guidance and for serving on my research proposal committee.

The research that appears in this thesis is the result of collaboration with incredible people: Omer Ben-Porat, Liu Leqi, Zachary C. Lipton, Michal Moshkovitz, Ulrike Schmidt-Kraepelin, and Eliran Shabat- I have learned a lot from each of you, and truly enjoyed working with you.

I had the pleasure of meeting many individuals who shared my work environment, lunches, and coffees during my Ph.D. journey. I am grateful for the experience and the opportunity to have crossed paths with each and every one of you. In particular, making it through grad school would not have been possible nor as half fun without the support and distraction of my good friends Esty Kelman, Noam Mazor, Uri Meir, and Shir Peleg. Thank you for helping me remain sane throughout the way, often with long discussions about topics outside your scope!

Finally, I'm incredibly grateful for my family. Thanks to my grandmother Tana, for the opportunities and unconditional love and support she has given me. To my beloved dogs Hunter (R.I.P) and Goliath for being the most amazing companions and friends in the world. Finally, I would like to thank my wonderful husband Oria, who is always there for me (even when it means escorting me across the world), always believing in me, and constantly inspires me to be a better person. You are the best choice I have ever made!

Abstract

Over the last decades, the field of Machine Learning (ML) has enjoyed tremendous success, wielding massive influence on our everyday lives with an apparent impact on a vast variety of applications such as recommender systems, autonomous vehicles, and machine translation. Computational Learning Theory is the very foundation for current and future machine learning methodologies. In this thesis, we expand and develop efficient ML methodologies in the areas of Online Learning, Generalization, and accommodating Societal Concerns such as Fairness.

The results of this thesis have been published in five research papers, out of which four have been published (in FORC'20, NeurIPS'20, NeurIPS'21, and AAAI'22).

Contents

Chapter 1

Introduction

Algorithms, particularly ones that are deployed for Artificial Intelligence (AI) systems, have a huge influence on our everyday lives. An algorithm is a set of data-based instructions that given any input returns some output. Netflix uses algorithms to map past show preferences into recommendations for movies we might be interested watching next, robotic vacuum cleaners use algorithms to clean our homes efficiently, algorithms are also responsible for more than 70 percent of US stock market trades, and soon will make autonomous cars drive us around.

AI offers several benefits that makes it a valuable tool for nearly any modern organization. It boosts the efficiency of products and services via capabilities such as personalized product recommendations. Services like auto translations and auto transcriptions are making the word much more accessible than ever. Using AI, a repetitive task can be performed automatically without humans feeling fatigued or bored by it. Data can be analyzed much faster and on a much larger scale than previously done by humans, enabling patterns to be found humans would never have seen otherwise. And finally, by harvesting and interpreting data, systems can be

trained to become more accurate than humans for crucial tasks like identifying or treating cancerous growths.

One of the main topics of my research is Machine learning (ML), a discipline that lies within the intersection of computer science, applied mathematics, and statistics. In recent decades, machine learning has grown increasingly important, and emerged as an important pillar of modern AI and data science research. ML is primarily concerned with extracting models from data and using these models to make accurate predictions.

Online Learning

Algorithms for online learning are especially challenging due to various constraints the learner needs to comply with: bounded computation time, handling unlimited amount of data, optimizing the performance throughout the entire learning phase (regret minimization), and up to the ability of adapting to a dynamic environment that depends on the actions of the learner (Reinforcement Learning).

Reinforcement Learning (RL) takes a step to a concerned with online learning where the goal of the system is to both learn the environment in terms of what are the consequences of each action and what is the ideal action in the current state. The goal of the learner in RL setting is either to find an optimal policy efficiently or to maximize the notion of cumulative reward. Common settings for RL problems include the Markov Decision Process (MDP) model, and the Multi-Arm Bandits (MAB) model.

Societal Challenges

As much as these advances improve the quality of our lives, as individuals and as a society, they come with a cost. It has been shown that algorithms can be exposed to adversarial attacks (e.g. against Autonomous Driving Models), risk our

private information (e.g., in 2007 researchers were able to identify individual users by matching the Netflix Prize data sets with film ratings on the Internet Movie Database[1]).

The second main topic of my research is societal challenges in the context of AI. One of the most important societal challenges is *Fairness*. Fairness emerges from having an unjustified bias against a subgroup (of population), described by a *protected feature* – race, gender, disability, etc.. When two different subgroups have a certain level of inequality in some measure – e.g., acceptance rate for a job/college regardless of the true ability of the candidates - we consider it discrimination (or "unfair"). In fairness research, we try to mitigate these biases by identifying when they occur and fix them using systematic tools. Fairness differs from other statistical biases in machine learning since it involves and affects people.

Another societal challenge that was studied in this thesis is safety, and in particular *safe RL*. Most research in reinforcement learning (RL) deals with the problem of learning an optimal policy for some Markov decision process (MDP). Safe RL focuses on finding the best policy that meets safety requirements. Typically, these problems are handled by adjusting the objective to include safety requirements and then optimizing over it, or incorporating additional safety constraints to the exploration stage. Instead, we addresses the safety of a specific MDP policy by detecting anomalous events rather than finding a policy that satisfies some pre-defined safety constraints (anomaly detection).

Addressing these vulnerabilities can help society to harness the full power of machine learning.

[1]Netflix prize privacy concern, retrieved from `http://en.wikipedia.org/wiki/Recommender_system\#Privacy_Concerns`

1.1 Thesis Contributions

This dissertation is focused around learning theory and societal challenges. In Chapter 2, we concentrate on Generalization in Machine Learning, where we start by showing how to derive Uniform Convergence guarantees for Multicalibration, a notion for group fairness. Uniform convergence of a hypothesis class refers to the property that the difference between the true error (risk) of each hypothesis and its estimate (empirical risk) approaches zero as the sample size approaches infinity. Uniform convergence is an important concept in machine learning because it provides guarantees on the accuracy of the learned model with respect to the true data distribution as well, which is essential for the model's generalization performance. In other words, if a model is learned using a hypothesis class that exhibits uniform convergence, it is more likely to generalize well to new, unseen data. Multicalibration of a predictor ensures that the predictor is calibrated across different (large enough) subpopulations. Selecting a multicalibrated predictor can prevent uncalibrated predictions for certain subgroups or populations, which can be problematic in many real-world applications, especially when the model's predictions have a significant impact on people's lives, such as in healthcare or finance. By ensuring that the model is equally accurate for all subgroups, multicalibration can help to avoid potential harms or biases in the model's predictions. In a work that was published in NeurIPS 2020 [117], we provide uniform convergence guarantees for multicalibration. More precisely, we derive sample complexity bounds to achieve uniform convergence for multicalibration. Our work focuses on addressing the issue of multicalibration error by separating it from the prediction error. Decoupling the fairness metric (multicalibration) from the accuracy (prediction error) is crucial due to the natural tradeoff between the two, and the societal decision on what constitutes an appropriate tradeoff, often mandated by regulators. We show the necessary sample

complexity for Predictor Classes with Finite Graph Dimension (these results appeared in a work which was published in [117]). We extend these results, showing how to use the same techniques to obtain Uniform Convergence for another notion commonly used in practice, F-Score. In addition, we improve the lower bounds and show a dependence on the size of the class (for finite predictor classes) and on the Graph dimension for infinite predictors.

Chapter 3 is devoted to results for the Online Learning regime. The first two sections (Sections 3.1 and 3.2), are in the MAB setting, a particular setting of RL. In general, MAB models decision-making problems in which an agent must choose between multiple actions or options, each with an unknown reward or payoff. The goal is to maximize the cumulative reward over a given time horizon, while simultaneously exploring the different actions to learn about their reward distributions. In Dueling bandits [133], the realization of the rewards is no longer the feedback. As an alternative, the learner chooses a pair of arms and the observed feedback is the winning arm between their "duel", the arm with the larger reward in the current round. The problem is motivated by web search optimization, where each action models a possible search result, and we are only given feedback regarding the preferred result. We refer the reader to [19] and [123] for surveys on dueling bandits. In Section 3.1, we generalize the dueling bandit setting to accommodate noisy comparisons of disjoint pairs of k-sized *teams* (subsets of arms) from a universe of n arms (players). These results were published in [42]. The problem is not only a generalization of an existing model in ML, but is also deeply connected to tournament solutions, that originated from social choice. Our framework formalizes a societal issue- learning about the "winning" team in team sports, where players cannot play for the same team simultaneously.

In Section 3.2, we accommodate another societal approach in MAB. We formalize

and address the dissatisfaction of users from a MAB-based recommender system, that may depart (and never come back). While naive approaches cannot handle this setting, we provide an efficient learning algorithm that achieves $\tilde{O}(\sqrt{T})$ regret, where T is the number of users. These results were published in [17].

In Section 3.3, the last part of the chapter, we assume a Markov model (a more general setting than MAB in RL), but there our goal is to find a characterization of a given policy that captures popular trajectories rather than finding the best policy. In most research in Safe RL, the focus is on finding the best policy that meets safety requirements. Typically, these problems are handled by adjusting the objective to include safety requirements and then optimizing over it, or incorporating additional safety constraints to the exploration stage. Anomaly Detection is the problem of identifying patterns in data that do not correspond to what is expected, i.e., anomalies. Anomaly Detection addresses a variety of applications: cyber-security, fraud detection, failure detection, etc. (see [32] for survey). This work takes an anomaly detection approach for safe RL and has implications for safety and explainability, both are societal challenges. As such, the model suggests a solution that is based on popular behavior and depends on society's needs rather than just satisfying pre-defined safety constraints. The results of this work appear in [41].

Finally, in Chapter 4 we present the results which have pure societal motivation. In particular, we address the problem of candidate screening in the multi-test setting, considering both Bernoulli and Gaussian models. We inspect the problem from a classic ML viewpoint (loss minimization), then we characterize the optimal policy when employees constitute a single group, demonstrating some interesting trade-offs. Subsequently, we address the multi-group setting, demonstrating fundamental impossibility results as well as optimal fairness solutions based on dynamic and group-depended decision rule. These results were published in [39].

Chapter 2

Generalization in ML

Introduction

Characterizing learnability is a fundamental problem in learning theory. For example, we already know that for supervised classification learnability is equivalent to uniform convergence. In addition, given a learnable problem and 'enough' data, any empirical risk minimization (ERM) algorithm would return an accurate predictor. There are many other interesting aspects of generalization. For example, in the following work we showed uniform convergence for multiclaibration of predictor classes with finite graph dimension. Multicalibration is a generalized notion of Calibration that provides a comprehensive methodology to address group fairness, where calibration is a common notion for ML tasks.

Sample Complexity of Uniform Convergence for Multicalibration In a work that was published in NeurIPS 2020 [117], we address the multicalibration error and decouple it from the prediction error. The importance of decoupling the fairness metric (multicalibration) and the accuracy (prediction error) is due to the inherent trade-off

between the two, and the societal decision regarding the "right tradeoff" (as imposed many times by regulators). Our work gives sample complexity bounds for uniform convergence guarantees of multicalibration error, which implies that regardless of the accuracy, we can guarantee that the empirical and (true) multicalibration errors are close. Our results are the first to apply for not only realizable settings, but also for agnostic settings. Agnostic and realizable are terms that describe different types of settings for a learning problem. In the realizable setting, we assume that there exists a function that perfectly fits the data, with no errors on the true distribution, and in particular, no errors on the training set, and the goal is to find it. In the agnostic setting, the learning algorithm does not make any assumptions about the relationship between the hypothesis class and the data distribution and only competes with the best predictor in the class, and the goal is to find a function that minimizes the generalization error, which is the error rate on unseen data. In general, this setting is more challenging because the algorithm cannot rely on prior knowledge. Moreover, realizable settings are less common in practice because the realizability assumption is often not realistic in real-world problems, especially in societal-related problems. Our results are also not restricted to a specific type of algorithm (such as deferentially private), and improve over previous multicalibration sample complexity bounds. Finally, they imply uniform convergence guarantees for the classical calibration error as well (i.e., not as a fairness notion). Our sample bounds guarantee uniform convergence for both finite predictor classes (logarithmic in the size of the predictor class) and infinite predictor classes, for which the bound depends on the *graph dimension* of the class, a measure of the class complexity and in particular an extension of the VC dimension for multiclass predictions. Finally, we derive a lower bound on the sample size required.

Our approach has the advantage of providing fairness "for free". Namely, whenever

the learner finds (e.g., by ERM) several predictors that minimize other desired constraints of her choice, she can simply select the predictor with the lowest estimated multicalibration error and improve fairness. As a result, an important tool of generalization from ML can now be paired together with one of the (current) major fairness notions.

In this thesis we also extend these results, showing how to use the same techniques to obtain Uniform Convergence for another notion commonly used in practice, F-Score. In addition, we improve the lower bounds and show a dependence on the size of the class (for finite predictor classes) and on the Graph dimension for infinite predictors.

2.1 Uniform Convergence for Multicalibration

2.1.1 Introduction

Data driven algorithms influence our everyday lives. While they introduce significant achievements in face recognition, to recommender systems and machine translation, they come at a price. When deployed for predicting outcomes that concern individuals, such as repaying a loan, surviving surgery, or skipping bail, predictive systems are prone to accuracy disparities between different social groups that often induce discriminatory results. These significant societal issues arise due to a variety of reasons: problematic analysis, unrepresentative data and even inherited biases against certain social groups due to historical prejudices. At a high level, there are two separate notions of fairness: *individual fairness* and *group fairness*. Individual fairness is aimed to guarantee fair prediction to each given individual, while group fairness aggregates statistics of certain subpopulations, and compares them. There is a variety of fairness notions for group fairness, such as demographic parity, equalized odds, equalized

opportunity, and more (see [12]). Our main focus would be on multicalibration criteria for group fairness [66]. Multicalibration of a predictor is defined as follows. There is a prespecified set of subpopulations of interest. The predictor returns a value for each individual (which can be interpreted as a probability). The multicalibration requires that for any "large" subpopulation, and for any value which is predicted "frequently" on that subpopulation, the predicted value and average realized values would be close on this subpopulation. Note that calibration addresses the relationship between the predicted and average realized values, and is generally unrelated to the prediction quality. For example, if a population is half positive and half negative, a predictor that predicts for every individual a value of 0.5 is perfectly calibrated but has poor accuracy. The work of [66] proposes a specific algorithm to find a multicalibrated predictor and derived its sample complexity. The work of [92] related the calibration error to the prediction loss, specifically, it bounds the calibration error as a function of the difference between the predictor loss and the Bayes optimal prediction loss. Their bound implies that in a realizable setting, where the Bayes optimal hypothesis is in the class, using ERM yields a vanishing calibration error, but in an agnostic setting this does not hold. With the motivation of fairness in mind, it is important to differentiate between the prediction loss and the calibration error. In many situations, the society (through regulators) might sacrifice prediction loss to improve fairness, and the right trade-off between them may be task dependent. On the other hand, calibration imposes self-consistency, namely, that predicted values and the average realized values should be similar for any protected group. In particular, there is no reason to prefer un-calibrated predictors over calibrated ones, assuming they have the same prediction loss. An important concept in this regard is uniform convergence. We would like to guarantee that the multicalibration error on the sample and the true multicalibration error are similar. This will allow society to rule-out un-calibrated

predictors when optimizing over accuracy and other objectives that might depend on the context and the regulator.

Our main results in this work are sample bounds that guarantee uniform convergence of a given class of predictors. We start by deriving a sample bound for the case of a finite hypothesis class, and derive a sample complexity bound which is logarithmic in the size of the hypothesis class. Later, for an infinite hypothesis class, we derive a sample bound that depends on the *graph dimension* of the class (which is an extension of the VC dimension for multiclass predictions). Finally, we derive a lower bound on the sample size required.

Technically, an important challenge in deriving the uniform convergence bounds is that the multicalibration error depends, not only on the correct labeling but also on the predictions by the hypothesis, similar in spirit to the internal regret notion in online learning. We remark that these techniques are suitable to reproduce generalization bounds for other complex measures such as F-score.

We stress that in contrast to previous works that either attained specific efficient algorithms for finding calibrated predictors [66] or provided tight connections between calibration error and prediction loss (mainly in the realizable case) [93], we take a different approach. We concentrate on the statistical aspects of generalization bounds rather than algorithmic ones, and similar to much of the generalization literature in machine learning derive generalization bounds over calibration error for *any* predictor class with a finite size or a finite graph dimension.

Nevertheless, our work does have algorithmic implications. For example, similarly to running ERM, running empirical multicalibration risk minimization over a hypothesis class with bounded complexity $\mathcal{H}$ and "large enough" training set, would output a nearly-multicalibrated predictor, assuming one exists. We guarantee that the empirical and true errors of this predictor would be similar, and derive the required sample size

either as a function of the logarithm of the size of the predictor class or of its finite graph dimension. Our bounds improve over previous sample complexity bounds and also apply in more general settings (e.g., agnostic learning). So while multicalibration uniform convergence is not formally necessary for learning multicalibrated predictors, the advantage of our approach is that the learner remains with the freedom to choose any optimization objectives or algorithms, and would still get a good estimation of the calibration error. To the best of our knowledge, this also introduces the first uniform convergence results w.r.t. calibration as a general notion (i.e., even not as a fairness notion).

Related work: Calibration has been extensively studied in machine learning, statistics and economics [57, 22, 56], and as a notion of fairness dates back to the 1960s [37]. More recently, the machine learning community adapted calibration as an anti-discrimination tool and studied it and the relationship between it and other fairness criteria [35, 44, 83, 109, 95]. There is a variety of fairness criteria, other than calibration, which address societal concerns that arise in machine learning. Fairness notions have two major categories. *Individual-fairness*, that are based on similarity metric between individuals and require that similar individuals will be treated similarly [47]. *Group-fairness*, such as demographic-parity and equalized-odds, are defined with respect to statistics of subpopulations [12]. Generalization and uniform convergence are well-explored topics in machine learning, and usually assume some sort of hypotheses class complexity measures, such as VC-dimension, Rademacher complexity, Graph-dimension and Natarajan-dimension [16, 45, 118]. In this work we build on these classic measures to derive our bounds. Generalization of fairness criteria is a topic that receives great attention recently. The works of [82, 132] define metric notions that are based on [47] and derive generalization guarantees. Other works relax the assumption of a known fairness metric and derive generalization

with respect to Individual Fairness based on oracle queries that simulate human judgments [59, 14, 71]. Bounds for alternative fairness notions, such as equalized-odds, gerrymandering, multi-accuracy, and envy-free appear in [130, 79, 81, 11]. We remark that this work does not provide generalization bounds for margin classifiers in the context of fairness, and we leave it for future work.

Multicalibration is a group-fairness notion that requires calibration to hold simultaneously on multiple subpopulations [66]. They proposed a polynomial-time differentially-private algorithm that learns a multicalibrated predictor from samples in agnostic setup. A byproduct of their choice of Differently Private algorithm is that their algorithm and analysis is limited to a finite domain. Our work provides generalization uniform convergence bounds that are independent of the algorithm that generates them, and also improve their sample bounds. The work of [93] bounds the calibration error by the square-root of the gap between its expected loss and the Bayes-optimal loss, for a broad class of loss functions. While in realizable settings this gap is vanishing, in agnostic settings this gap can be substantial. Our results do not depend on the hypothesis' loss to bound the calibration error, which allows us to give guarantees in the agnostic settings as well.

2.1.2 Model and Preliminaries

Let $\mathcal{X}$ be any finite or countable domain (i.e., $\mathcal{X}$ is a population and each domain point encodes an individual) and let $\{0,1\}$ be the set of possible *outcomes*. Let D be a probability distribution over $\mathcal{X} \times \{0,1\}$, i.e., a joint distribution over domain points and their outcomes. Intuitively, given pairs (x_i, y_i), we assume that outcomes $y_i \in \{0,1\}$ are the realizations of underlying random sampling from independent Bernoulli distributions with (unknown) parameters $p^*(x_i) \in [0,1]$. The goal of the learner is to predict the (unknown) parameters $p^*(x_i)$, given a domain point x_i. Let

$\mathcal{Y} \subseteq [0,1]$ be the set of possible *predictions values*. A *predictor* (hypothesis) h is a function that maps domain points from $\mathcal{X}$ to prediction values $v \in \mathcal{Y}$. A set of predictors $h : \mathcal{X} \to \mathcal{Y}$ is a *predictor class* and denoted by $\mathcal{H}$. Let $\Gamma = \{U_1, ..., U_{|\Gamma|}\}$ be a finite collection of subpopulations (possibly overlapping) from the domain $\mathcal{X}$ (technically, Γ is a collection of subsets of $\mathcal{X}$). Throughout this chapter, we will distinguish between the case where $\mathcal{Y}$ is a finite subset of $[0,1]$ and the case where $\mathcal{Y} = [0,1]$ (continuous). Both cases depart from the classical binary settings where $\mathcal{Y} = \{0,1\}$, as predictors can return any prediction value $v \in \mathcal{Y}$ (e.g., $v = 0.3$). We define Λ to be a *partition* of $\mathcal{Y}$ into a finite number of subsets, that would have different representations in the continuous and finite cases. For the continuous case where $\mathcal{Y} = [0,1]$, we would partition $\mathcal{Y}$ into a finite set of intervals using a *partition parameter* $\lambda \in (0,1]$ that would determinate the lengths of the intervals. Namely, $\Lambda_\lambda := \{\{I_j\}_{j=0}^{\frac{1}{\lambda}-1}\}$, where $I_j = [j\lambda, (j+1)\lambda)$. When $\mathcal{Y}$ is finite, Λ would be a set of singletons: $\Lambda = \{\{v\} : v \in \mathcal{Y}\}$ and $h(x) \in I = \{v\}$ is equivalent to $h(x) = v$.

Definition 2.1 (Calibration error)**.** *The* calibration error *of predictor $h \in \mathcal{H}$ w.r.t. a subpopulation $U \in \Gamma$ and an interval $I \subseteq [0,1]$, denoted by $c(h, U, I)$ is the difference between the expectations of y and $h(x)$, conditioned on domain points from U that h maps to values in I. I.e.,*

$$c(h, U, I) := \mathop{\mathbb{E}}_{D}[y \mid x \in U, h(x) \in I] - \mathop{\mathbb{E}}_{D}[h(x) \mid x \in U, h(x) \in I]$$

Notice that for the case where $\mathcal{Y}$ is finite, we can rewrite the expected calibration error as

$$c(h, U, I = \{v\}) = \mathop{\mathbb{E}}_{D}[y \mid x \in U, h(x) = v] - v$$

Since calibration error of predictors is a measure with respect to a specific pair of

subpopulation U and an interval I, we would like to have a notion that captures "well-calibrated" predictors on "large enough" subpopulations and "significant enough" intervals I that h maps domain points (individuals) to, as formalized in the following definition.

Definition 2.2 (Category)**.** *A* category *is a pair (U, I) of a subpopulation $U \in \Gamma$ and an interval $I \in \Lambda$. We say that a category (U, I) is* interesting *according to predictor h and parameters $\gamma, \psi \in (0, 1]$, if $\Pr_D[x \in U] \geq \gamma$ and $\Pr_D[h(x) \in I \mid x \in U] \geq \psi$.*

We focus on predictors with calibration error of at most α for any interesting category.

Definition 2.3 ((α, γ, ψ)–multicalibrated predictor)**.** *A predictor $h \in \mathcal{H}$ is* (α, γ, ψ)–multicalibrated*, if for every interesting category (U, I) according to h, γ and ψ, the absolute value of the calibration error of h w.r.t. the category (U, I) is at most α, i.e., $\left|c(h, U, I)\right| \leq \alpha$.*

We define empirical versions for calibration error and (α, γ, ψ)–multicalibrated predictor.

Definition 2.4 (Empirical Calibration error)**.** *Let (U, I) be a category and let $S^m = \{(x_1, y_1), ..., (x_m, y_m)\}$ be a training set of m samples drawn i.i.d. from D. The* empirical calibration error *of a predictor $h \in \mathcal{H}$ w.r.t. (U, I) and S is:*

$$\hat{c}(h, U, I, S) := \sum_{i=1}^{m} \frac{\mathbb{I}\left[x_i \in U, h(x_i) \in I\right]}{\sum_{j=1}^{m} \mathbb{I}\left[x_j \in U, h(x_j) \in I\right]} y_i - \sum_{i=1}^{m} \frac{\mathbb{I}\left[x_i \in U, h(x_i) \in I\right]}{\sum_{j=1}^{m} \mathbb{I}\left[x_j \in U, h(x_j) \in I\right]} h(x_i),$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Notice that when $\mathcal{Y}$ is finite, since $h(x) \in \{v\}$ is equivalent to $h(x) = v$, we can re-write the empirical calibration error as: $\hat{c}(h, U, I = \{v\}, S) := \sum_{i=1}^{m} \frac{\mathbb{I}[x_i \in U, h(x_i) = v]}{\sum_{j=1}^{m} \mathbb{I}[x_j \in U, h(x_j) = v]} y_i - v$.

Definition 2.5 ((α, γ, ψ)–Empirically multicalibrated predictor)**.** *A predictor $h \in \mathcal{H}$ is* (α, γ, ψ)–empirically multicalibrated *on a sample S of i.i.d examples from D, if for every interesting category (U, I) according to h, γ and ψ, we have $\left|\hat{c}(h, U, I, S)\right| \leq \alpha$.*

We assume that the predictors are taken from some predictor class $\mathcal{H}$. Our main goal is to derive sample bounds for the empirical calibration error to "generalize well" for every $h \in \mathcal{H}$ and every interesting category. We formalize it as follows.

Definition 2.6 (Multicalibration Uniform Convergence)**.** *A predictor class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ has the* multicalibration uniform convergence *property (w.r.t. collection Γ) if there exist a function $m^{mc}_{\mathcal{H}}(\epsilon, \delta, \gamma, \psi) \in \mathbb{N}$, for $\epsilon, \delta, \gamma, \psi \in (0, 1]$, such that for every distribution D over $\mathcal{X} \times \{0, 1\}$, if $S^m = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ is a training set of $m \geq m^{mc}_{\mathcal{H}}(\epsilon, \delta, \gamma, \psi)$ examples drawn i.i.d. from D, then for every $h \in \mathcal{H}$ and every interesting category (U, I) according to h, γ and ψ, the difference between the calibration error and the empirical calibration error is at most ϵ with probability of at least $1 - \delta$, i.e., $\Pr_D[|\hat{c}(h, U, I, S^m) - c(h, U, I)| \leq \epsilon] > 1 - \delta$.*

We emphasize that the property of multicalibration uniform convergence w.r.t. a predictor class $\mathcal{H}$ is neither a necessary nor sufficient for having multicalibrated predictors $h \in \mathcal{H}$. Namely, having uniform convergence property implies only that the empirical and true errors are similar, but does not imply that they are small. In addition, having a predictor with zero multicalibration error (realizability) does not imply anything about the generalization multicalibration error. For example, if $\mathcal{H}$ contains all the possible predictors, there will clearly be a zero empirical error predictor who's true multicalibration error is very high.

When $\mathcal{H}$ is an infinite predictor class, we can achieve generalization by assuming a finite complexity measure. VC-dimension (the definition appears in Section 2.1.6) measures the complexity of binary hypothesis classes. In this work, we rephrase the generalization problem of multicalibration in terms of multiple generalization

problems of binary hypothesis classes with finite VC-dimension, and derive sample complexity bounds for it. So our goal is to approximate the (true) calibration error by estimating it on a large sample. Namely, we would like have a property which indicates that a large-enough sample will result a good approximation of the calibration-error for any hypothesis $h \in \mathcal{H}$ and any interesting category (U, I) according to h. Our technique for achieving this property uses known results about binary classification. We mention the definitions of "risk function", "empirical-risk function" and "uniform convergence for statistical learning" (the latter appears in Section 2.1.6). For this purpose, $h : \mathcal{X} \to \{0, 1\}$ would denote a binary hypothesis, $\ell : \mathcal{Y} \times \{0, 1\} \to \mathbb{R}_+$, denotes a loss function and D stays a distribution over $\mathcal{X} \times \{0, 1\}$.

Definition 2.7 (Risk function, Empirical risk)**.** *The* risk function*, denoted by L_D, is the expected loss of a hypothesis h w.r.t D, i.e., $L_D(h) := \mathbb{E}_{(x,y)\sim D}[\ell(h(x), y)]$. Given a random sample $S = ((x_i, y_i))_{i=1}^m$ of m examples drawn i.i.d. from D, the* empirical risk *is the average loss of h over the sample S i.e., $L_S(h) := \frac{1}{m}\sum_{i=1}^m \ell(h(x_i), y_i)$.*

Note that the definitions of uniform convergence for statistical learning and the multicalibration uniform convergence are distinct. A major difference is that while the notion of uniform convergence for statistical learning imposes a requirement on the risk, which is defined using an expectation over a fixed underlying distribution D, the notion of multicalibration uniform convergence imposes a requirement on the calibration error, in which the expectation is over a conditional distribution that depends on the predictor. When the prediction range, $\mathcal{Y}$, is discrete, we consider the standard multiclass complexity notions– Graph-dimension and Natarjan dimension, which are define as follows.

Definition 2.8 (Graph Dimension)**.** *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class from domain $\mathcal{X}$ to a finite set $\mathcal{Y}$ and let $S \subseteq \mathcal{X}$. We say that $\mathcal{H}$ G-shatters S if there exists a function $f : S \to \mathcal{Y}$ such that for every $T \subseteq S$ there exists a hypothesis $h \in \mathcal{H}$ such*

that $\forall x \in S : h(x) = f(x) \iff x \in T$. *The graph dimension of* $\mathcal{H}$, *denoted* $d_G(\mathcal{H})$, *is the maximal cardinality of a set that is G-shattered by* $\mathcal{H}$.

Definition 2.9 (Natarajan Dimension)**.** *Let* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ *be a hypothesis class from domain* $\mathcal{X}$ *to a finite set* $\mathcal{Y}$ *and let* $S \subseteq \mathcal{X}$. *We say that* $\mathcal{H}$ *N-shatters* S *if there exists functions* $f_0, f_1 : S \to \mathcal{Y}$ *such that for every* $x \subseteq S$ *it holds that* $f_0(x) \neq f_1(x)$, *and for every* $T \subseteq S$ *there exists a hypothesis* $h \in \mathcal{H}$ *such that* $\forall x \in T : h(x) = f_0(x)$, *and* $\forall x \in S \setminus T, h(x) = f_1(x)$. *The graph dimension of* $\mathcal{H}$, *denoted* $d_G(\mathcal{H})$, *is the maximal cardinality of a set that is N-shattered by* $\mathcal{H}$.

Our Contributions

Within [117], we derived two upper bounds. The first one (in Theorem 2.10) is for finite predictor classes, in which we discretize $\mathcal{Y} = [0, 1]$ into Λ_λ and derive a bound which depends logarithmicly on λ^{-1}. The second one (Theorem 2.11) is for infinite predictor classes with discrete prediction values set $\mathcal{Y}$ and finite graph-dimension. We also complemented the upper bounds with a lower bound result in Theorem 2.12. In this thesis we improve the lower bound, making it...[CONTINUE FROM HERE[]]

Theorem 2.10. *Let* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ *be a finite predictor class. Then,* $\mathcal{H}$ *has the uniform multicalibration convergence property with* $m^{mc}_{\mathcal{H}}(\epsilon, \delta, \gamma, \psi) = O\left(\frac{1}{\epsilon^2\gamma\psi} \log\left(|\Gamma||\mathcal{H}|/\delta\lambda\right)\right)$.

Theorem 2.11. *Let* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ *be an infinite predictor class from domain* $\mathcal{X}$ *to a discrete prediction set* $\mathcal{Y}$ *with finite graph-dimension* $d_G(\mathcal{H}) \leq d$, *then* $\mathcal{H}$ *has the uniform multicalibration convergence property with* $m^{mc}_{\mathcal{H}}(\epsilon, \delta, \gamma, \psi) = O\left(\frac{1}{\epsilon^2\psi^2\gamma}\left(d + \log\left(|\Gamma||\mathcal{Y}|/\delta\right)\right)\right)$.

Theorem 2.12. *Let* $\mathcal{H}$ *be a finite predictor class or an infinite predictor class with finite graph-dimension* $d_G(\mathcal{H}) \leq d$. *Then,* $\mathcal{H}$ *has multicalibration uniform convergence with* $m(\epsilon, \delta, \psi, \gamma) = \Omega(\frac{1}{\psi\gamma\epsilon^2} \ln(1/\delta))$ *samples.*

Relation to prior work

Rewriting the sample bound of [66] using our parameters, they have a sample complexity of $O\left(\frac{1}{\epsilon^3\psi^{3/2}\gamma^{3/2}}\log(\frac{|\Gamma|}{\epsilon\gamma\delta})\right)$. Comparing the bounds, the most important difference is the dependency on ϵ, the generalization error. They have a dependency of ϵ^{-3}, while we have of ϵ^{-2}, which is tight due to our lower bound. For the dependency on γ, they have $\gamma^{-3/2}$, while we have γ^{-1}, which is also tight. For the dependency on ψ, they have $\psi^{-3/2}$, while we have ψ^{-1} for a finite hypothesis class (which is tight due to our lower bound) and ψ^{-2} for an infinite hypothesis class. Finally, recall that the bound of [66] applies only to their algorithm and since it is a differentially private algorithm, it requires the domain $\mathcal{X}$ to be finite, while our results apply to continuous domains as well. Note that having (α, γ, ψ)– empirically multicalibrated predictor on large random sample, guarantees that, with high probability, it is also $(\alpha + \epsilon, \gamma, \psi)$–mutlicalibrated with respect to the underlying distribution, where ϵ is the generalization error that depends on the sample size.

2.1.3 Predictor Classes with Finite Graph Dimension

fThroughout this section we assume that the predictions set $\mathcal{Y}$ is discrete. This assumption allows us to analyze the multicalibration generalization of possibly infinite hypothesis classes with finite known multiclass complexity measures such as the graph-dimension. (We discuss the case of $\mathcal{Y} = [0, 1]$ at the end of the section.) Recall that in this setup, the prediction-intervals set, Λ, contains singleton intervals with values taken from $\mathcal{Y}$, namely, $\Lambda = \{\{v\} \mid v \in \mathcal{Y}\}$. Thus, if a prediction, $h(x)$ is in the interval $\{v\}$, it means the prediction value is exactly v, i.e., $h(x) \in \{v\} \Leftrightarrow h(x) = v$. As we have mentioned earlier, part of our technique is to reduce multicalibration generalization to the generalization analysis of multiple binary hypothesis classes to get sample complexity bounds. The Fundamental Theorem of Statistical Learning

(see Theorem 2.33, Section 2.1.6) provides tight sample complexity bounds for uniform convergence for binary hypothesis classes. A direct corollary of this theorem indicates that by using "large enough" sample, the difference between the true probability to receive a positive outcome and the estimated proportion of positive outcomes, is small, with high probability.

Corollary 2.13. *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a binary hypothesis class with $VCdim(\mathcal{H}) \leq d$. Then, there exists a constant $C \in \mathbb{R}$ such that for any distribution D, and parameters $\epsilon, \delta \in (0,1)$, if $S = \{x_i, y_i\}_{i=1}^m$ is a sample of m i.i.d. examples from D, and $m \geq C((d + \log(1/\delta))/\epsilon^2)$ then with probability at least $1-\delta$, $\forall h \in \mathcal{H}$: $\left|\frac{1}{m}\sum_{i=1}^m h(x_i) - \Pr_{x\sim D}[h(x)=1]\right| < \epsilon$.*

Before we move on, we want to emphasize the main technical challenge in deriving generalization bounds for infinite predictor classes. Unlike PAC learning, in multi-calibration learning the distribution over the domain is dependent on the predictors class. Each pair of $h \in \mathcal{H}, v \in \mathcal{Y}$ induce a distribution over the domain points x such that $h(x) = v$. As the number of predictors in the class is infinite, we cannot apply a simple union bound over the various induced distributions. This is a main challenge in our proof. In order to utilize the existing theory about binary hypothesis classes we have to represent the calibration error in terms of binary predictors. For this purpose, we define the notion of "binary predictor class", $\mathcal{H}_v \subseteq \{0,1\}^{\mathcal{X}}$, that depends on the original predictor class $\mathcal{H}$ and on a given prediction value $v \in \mathcal{Y}$. Each binary predictor $h_v \in \mathcal{H}_v$ corresponds to a predictor $h \in \mathcal{H}$ and value $v \in \mathcal{Y}$ and predicts 1 on domain points x if h predicts v on them (and 0 otherwise).

Definition 2.14 (Binary Predictor)**.** *Let $h \in \mathcal{H}$ be a predictor and let $v \in \mathcal{Y}$ be a prediction value. The* binary predictor *of h and v, denoted $h_v(x)$, is the binary function that receives $x \in \mathcal{X}$ and outputs 1 iff $h(x) = v$, i.e., $h_v(x) = \mathbb{I}[h(x) = v]$.*

The binary predictor class *w.r.t. the original predictor class $\mathcal{H}$ and value $v \in \mathcal{Y}$,*

denoted by $\mathcal{H}_v$, is defined as $\mathcal{H}_v = \{h_v : h \in \mathcal{H}\}$.

The definition of binary predictors alone is not sufficient since it ignores the outcomes $y \in \{0, 1\}$. Thus, we define true positive function, $\phi_{h_v} \in \Phi_{\mathcal{H}_v}$, that corresponds to a binary predictor h_v, such that given a pair $(x \in \mathcal{X}, y \in \{0, 1\})$, it outputs 1 iff $h_v(x) = 1$ and $y = 1$.

Definition 2.15 (True positive function)**.** *Let $\mathcal{H}_v \subseteq \{0, 1\}^{\mathcal{X}}$ be a binary predictor class and let $h_v \in \mathcal{H}_v$ be a binary predictor. Then, the true positive function w.r.t. h_v is $\phi_{h_v}(x, y) := \mathbb{I}[h_v(x) = 1, y = 1]$. The* true positive class *of $\mathcal{H}_v$, is defined $\Phi_{\mathcal{H}_v} := \{\phi_{h_v} : h_v \in \mathcal{H}_v\}$.*

Using the above definitions we can re-write the calibration error as follows. Let $I_v = \{v\}$ be a singleton interval. Then, the calibration error and the empirical calibration errors take the following forms:

$$c(h, U, I_v) = \mathop{\mathbb{E}}_{D}[y \mid x \in U, h(x) = v] - v = \Pr_{(x,y)\sim D}[y = 1 \mid x \in U, h(x) = v] - v.$$

$$\begin{aligned}\hat{c}(h, U, I_v, S) &= \sum_{i=1}^{m} \frac{\mathbb{I}[x_i \in U, h(x_i) = v]}{\sum_{j=1}^{m} \mathbb{I}[x_j \in U, h(x_j) = v]} y_i - v \\ &= \frac{\sum_{i=1}^{m} \mathbb{I}[x_i \in U, h(x_i) = v, y_i = 1]}{\sum_{j=1}^{m} \mathbb{I}[x_j \in U, h(x_j) = v]} - v.\end{aligned}$$

The probability term in the calibration error notion is conditional on the subpopulation $U \in \Gamma$ and on the prediction value $h(x)$. Thus, different subpopulations and different predictors induce different distributions on the domain $\mathcal{X}$. To understand the challenge, consider the collection of conditional distributions induced by $h \in \mathcal{H}$ and an interesting category (U, I). Since $\mathcal{H}$ is infinite, we have an infinite collection of distributions, and guaranteeing uniform convergence for such a family of distributions

is challenging. In order to use the fundamental theorem of learning (Theorem 2.33), we circumvent this difficulty by re-writing the calibration error as follows.

$$c(h, U, I_v) = \Pr_{(x,y)\sim D}[y = 1 \mid x \in U, h(x) = v] - v = \frac{\Pr[y = 1, h(x) = v \mid x \in U]}{\Pr[h(x) = v \mid x \in U]} - v.$$

Later , we will separately approximate the numerator and denominator.
Finally, we use the definitions of binary predictor, h_v, and true positive functions ϕ_{h_v}, to represent the calibration error in terms of binary functions. Thus, the calibration error and the empirical calibration error take the following forms:

$$c(h, U, I_v) = \frac{\Pr[y = 1, h(x) = v \mid x \in U]}{\Pr[h(x) = v \mid x \in U]} - v = \frac{\Pr[\phi_{h_v}(x, y) = 1 \mid x \in U]}{\Pr[h_v(x) = 1 \mid x \in U]} - v,$$

$$\begin{aligned}\hat{c}(h, U, I_v, S) &= \frac{\sum_{i=1}^{m} \mathbb{I}[x_i \in U, h(x_i) = v, y_i = 1]}{\sum_{j=1}^{m} \mathbb{I}[x_j \in U, h(x_j) = v]} - v \\ &= \frac{\sum_{i=1}^{m} \mathbb{I}[x_i \in U, \phi_{h_v}(x_i, y_i) = 1]}{\sum_{j=1}^{m} \mathbb{I}[x_j \in U, h_v(x_j) = 1]} - v.\end{aligned}$$

Since the calibration error as written above depends on binary predictors, if we can prove that the complexity of the hypothesis classes containing them has finite VC-dimension, then we will be able to approximate for each term separately. Recall that in this section we are dealing with multiclass predictors, which means that we must use multiclass complexity notion. We analyze the generalization of calibration by assuming that the predictor class $\mathcal{H}$ has a finite graph-dimension. The following lemma states that a finite graph dimension of $\mathcal{H}$ implies finite VC-dimension of the binary prediction classes $\mathcal{H}_v$ for any $v \in \mathcal{Y}$. This result guarantees good approximation for the denominator term, $\Pr[h_v(x) = 1 \mid x \in U]$, in the calibration error. We remark that while the following lemma is also a direct corollary when considering graph dimension as a special case of Psi-dimension [16], for completeness,

Lemma 2.16. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a predictor class such that $d_G(\mathcal{H}) \leq d$. Then, for any $v \in \mathcal{Y}$, $VCdim(\mathcal{H}_v) \leq d$.*

In addition to the complexity bound of the binary predictor classes $\mathcal{H}_v$, we would like to derive a bound on the VC-dimension of the prediction-outcome classes $\Phi_{\mathcal{H}_v}$ which would enable a good approximation of the numerator term, $\Pr[\phi_{h_v}(x, y) = 1 \mid x \in U]$ in the calibration error. This bound is achieved by using the following lemma that indicates that the VC-dimension of $\Phi_{\mathcal{H}_v}$ is bounded by the VC-dimension of $\mathcal{H}_v$.

Lemma 2.17. *Let $\mathcal{H}_v \subseteq \{0, 1\}^{\mathcal{X}}$ be a binary predictor class with $VCdim(\mathcal{H}_v) \leq d$, and let $\Phi_{\mathcal{H}_v}$ be the true positive class w.r.t. $\mathcal{H}_v$. Then, $VCdim(\Phi_{\mathcal{H}_v}) \leq d$.*

The fact that the VC-dimensions of $\mathcal{H}_v$ and $\Phi_{\mathcal{H}_v}$ are bounded enables to utilize the existing theory and derive sampling bounds for accurate approximations for the numerator and the denominator of the calibration error with high probability, respectively. Lemma 2.18 formalizes these ideas.

Lemma 2.18. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a predictor class with $d_G(\mathcal{H}) \leq d$. Let $v \in \mathcal{Y}$ be a prediction value and let $U \subset \mathcal{X}$ be a subpopulation. Then, there exist a constant $C \in \mathbb{R}$ such that for any distribution D over $\mathcal{X} \times \{0, 1\}$ and $\epsilon, \delta \in (0, 1)$, if D_U is the induced distribution on $U \times \{0, 1\}$ and $S = \{x_i, y_i\}_{i=1}^m$ is a random sample of size $m \geq C\frac{d+\log(1/\delta)}{\epsilon^2}$ drawn i.i.d. according to D_U, then with probability at least $1 - \delta$ for every $h \in \mathcal{H}$:*

$$\left|\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left[h(x_i) = v\right] - \Pr_{D_U}[h(x) = v]\right| \leq \epsilon, \quad \textit{and}$$
$$\left|\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left[h(x_i) = v, y = 1\right] - \Pr_{D_U}[h(x) = 1, y = 1]\right| \leq \epsilon.$$

Having an accurate approximation of the denominator and numerator terms of the calibration error does not automatically implies good approximation for it. For

example, any approximation error in the numerator is scaled by 1 divided by the denominator's value. The following lemma tells us how accurate the approximations of the numerator and the denominator should be in order to achieve good approximation of the entire fraction, given a lower bound on the true value of the denominator.

Lemma 2.19. *Let* $p_1, p_2, \tilde{p}_1, \tilde{p}_2, \epsilon, \psi \in [0,1]$ *such that* $p_1, \psi \leq p_2$ *and* $|p_1 - \tilde{p}_1|$, $|p_2 - \tilde{p}_2| \leq \psi\epsilon/3$. $|p_1/p_2 - \tilde{p}_1/\tilde{p}_2| \leq \epsilon$.

Since multicalibration uniform convergence requires empirical calibration errors of interesting categories to be close to their respective (true) calibration errors, a necessary condition is to have a large sample from every large subpopulation $U \in \Gamma$. The following lemma indicates the sufficient sample size to achieve a large subsample from every large subpopulation with high probability.

Lemma 2.20. *Let* $\gamma \in (0,1)$ *and let* $\Gamma_\gamma = \{U \in \Gamma \mid \Pr_{x\sim D}[x \in U] \geq \gamma\}$ *be the collection of subpopulations from* Γ *that has probability at least* γ *according to* D. *Let* $\delta \in (0,1)$ *and let* $S = \{(x_i, y_i)\}_{i=1}^m$ *be a random sample of* m *i.i.d. examples from* D. *Then, with probability at least* $1-\delta$, *if* $m \geq \frac{8}{\gamma}\log(|\Gamma|/\delta)$, *it holds that* $\forall U \in \Gamma_\gamma : |S \cap U| > \frac{\gamma m}{2}$.

The following theorem combines all the intuition described above and prove an upper bound on the sample size needed to achieve multicalibration uniform convergence. It assumes that the predictor class $\mathcal{H}$ has a finite graph-dimension, $d_G(\mathcal{H})$ and uses Lemma 2.16 and Lemma 2.17 to derive an upper bound on the VC-dimension of $\mathcal{H}_v$ and $\Phi_{\mathcal{H}_v}$. Then, it uses Lemma 2.18 to bound the sample complexity for "good" approximation of the numerator and the denominator of the calibration error.

Theorem 2.11. *Let* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ *be an infinite predictor class from domain* $\mathcal{X}$ *to a discrete prediction set* $\mathcal{Y}$ *with finite graph-dimension* $d_G(\mathcal{H}) \leq d$, *then* $\mathcal{H}$ *has the uniform multicalibration convergence property with* $m_{\mathcal{H}}^{mc}(\epsilon, \delta, \gamma, \psi) = O\left(\frac{1}{\epsilon^2\psi^2\gamma}\left(d + \log\left(|\Gamma||\mathcal{Y}|/\delta\right)\right)\right)$.

The proof of Theorem 2.11 uses the relative Chernoff bound (Lemma 2.30) to show that with probability at least $1-\delta/2$, every subpopulation $U \in \Gamma$ with $\Pr_D[U] \geq \gamma$, has a sub-sample of size at least $\frac{\gamma m}{2}$, namely $|S \cap U| \geq \frac{\gamma m}{2}$. Then, it uses Lemmas 2.16 and 2.17 to show that for every $v \in \mathcal{Y}$, $VCdim(\Phi_{\mathcal{H}_v}) \leq VCdim(\mathcal{H}_v) \leq d_G(\mathcal{H})$. It proceeds by applying Lemma 2.18 to show that, with probability at least $1-\delta/2$, for every prediction value $v \in \mathcal{Y}$ and every subpopulation $U \in \Gamma$, if $|S \cap U| \geq \frac{\gamma m}{2}$, then: $\left|\Pr\left[\phi_{h_v}(x,y)=1 \mid x \in U\right] - \frac{1}{|S \cap U|}\sum_{i=1}^{m} \mathbb{I}\left[x_i \in U, \phi_{h_v}(x_i,y_i)=1\right]\right| \leq \frac{\psi\epsilon}{3}$, and $\left|\Pr\left[h_v(x)=1 \mid x \in U\right] - \frac{1}{|S \cap U|}\sum_{j=1}^{m} \mathbb{I}\left[x_j \in U, h_v(x_j)=1\right]\right| \leq \frac{\psi\epsilon}{3}$.
Finally, it concludes the proof of Theorem 2.11 using Lemma 2.19.

The following corollary indicates that having (α,γ,ψ)– empirically multicalibrated predictor on a large random sample guarantees a $(\alpha+\epsilon,\gamma,\psi)$–mutlicalibrated predictor with respect to the underlying distribution with high probability, where ϵ is a generalization error that depends on the sample size. It follows immediately from Theorem 2.11.

Corollary 2.21. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a predictor class with $d_G(\mathcal{H}) \leq d$ and let D be a distribution over $\mathcal{X} \times \{0,1\}$. Let S be a random sample of m examples drawn i.i.d. from D and let $h \in \mathcal{H}$ be (α,γ,ψ)– empirically multicalibrated predictor on S. Then, there exists a constant $C>0$ such that for any $\epsilon,\delta \in (0,1)$, if $m \geq \frac{C}{\epsilon^2\psi^2\gamma}\left(d+\log\left(|\Gamma||\mathcal{Y}|/\delta\right)\right)$, then with probability at least $1-\delta$, h is $(\alpha+\epsilon,\gamma,\psi)$–multicalibrated w.r.t. the underlying distribution D.*

Finite versus continuous $\mathcal{Y}$: We have presented all the results for the infinite predictor class using a finite prediction-interval set $\Lambda = \{\{v\} | v \in \mathcal{Y}\}$. We can extend our results to the continuous $\mathcal{Y} = [0,1]$ in a straightforward way. We can simply round the predictions to a value $j\lambda$, and there are $1/\lambda$ such values. This will result in an increase in the calibration error of at most λ. (Note that in the finite predictor

class case, we have a more refine analysis that does not increase the calibration error by λ.) The main issue with this approach is that the graph-dimension depends on the parameter λ through the induced values $j\lambda$. Since we select λ and the points $j\lambda$, the magnitude of graph-dimension depends not only on the predictor class but also on parameters which are in our control, and therefore harder to interpret. For this reason we preferred to present our results for the finite $\mathcal{Y}$ case, and remark that one can extend them to the continuous $\mathcal{Y} = [0, 1]$ case.

2.1.4 Lower Bounds

We prove a lower bound for the required number of samples to get multicalibration uniform convergence. The proof is done by considering a predictor class with a single predictor that maps $\gamma\psi$ fraction of the population to $1/2 + \epsilon$. We show that this class has multicalibration uniform convergence property for $1/2 + \epsilon$ and then show how to use this property to distinguish between biased coins, which yield a lower bound of $\Omega(\frac{1}{\psi\gamma\epsilon^2}\ln(1/\delta))$ on the sample complexity.

Theorem 2.12. *Let $\mathcal{H}$ be a finite predictor class or an infinite predictor class with finite graph-dimension $d_G(\mathcal{H}) \leq d$. Then, $\mathcal{H}$ has multicalibration uniform convergence with $m(\epsilon, \delta, \psi, \gamma) = \Omega(\frac{1}{\psi\gamma\epsilon^2}\ln(1/\delta))$ samples.*

Theorem 2.22. *Let $\mathcal{H}$ be an infinite predictor class with finite graph-dimension, $d_N(\mathcal{H}) = d$. Then, for every $\psi \leq 1/2$, $\mathcal{H}$ has $(\epsilon, \psi, \gamma, \delta)-$multicalibration uniform convergence with $m = \Omega(d)$ samples.*

Theorem 2.23. *Let $\mathcal{H}$ be a finite predictor class or an infinite predictor class. Then, for every $\psi \leq 1/2$, $\mathcal{H}$ has $(\epsilon, \psi, \gamma, \delta)-$multicalibration uniform convergence with $m = \Omega(\log|\Gamma|)$ samples.*

2.1.5 F-Score Uniform Convergence

Given a binary classification task, F-score is a measure used in data science for the accuracy of a predictor. The F-score is a function of the precision and recall of the predictor, where precision is the number of true positive results divided by the amount of all (including mislabeled) positive samples, and recall is the number of true positive samples divided by the amount of all (truly) positive samples.

Formally, we define the following error measurements: true positive rate, precision, and recall. We then use these measurements in the formal definition of the $F-$scores.

Error measurements Let h be a predictor, and let D be a distribution over $\mathcal{X} \times \mathcal{Y}$. Then, the true positive (TP) of h is defined as

$$TP(h) := \Pr_{(x,y)\sim D}[h(x) = y = 1],$$

The precision and recall of a predictor h are defined as

$$precision(h) = \frac{TP(h)}{\mathbb{E}_{(x,y)\sim D}[h(x) = 1]} \quad recall(h) = \frac{TP(h)}{\mathbb{E}_{(x,y)\sim D}[y]}.$$

For convince, whenever it is clear from the context we write $\Pr[\cdot]$, instead of $\Pr_{(x,y)\sim D}[\cdot]$.

Moving on to the definitions of F_β-Score and F_1 scores.

Definition 2.24. *Let $h \in \{0,1\}^{\mathcal{X}}$ be a predictor, D denote be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\beta \geq 0$ be a parameter. The F_β score of h is*

$$F_\beta(h) = (1+\beta^2)\frac{precision(h) \cdot recall(h)}{\beta^2 \cdot precision(h)) + recall(h)} = \frac{(1+\beta^2)\Pr[h(x) = y = 1]}{\beta^2 \Pr[y = 1] + \Pr[h(x) = 1]}.$$

F_1 score is the balanced F_β-score, i.e.,

$$\begin{aligned} F_1(h) &= \frac{2}{precision^{-1}(h) + recall^{-1}(h)} \\ &= \frac{2\Pr[h(x) = y = 1]}{\Pr[y = 1] + \Pr[h(x) = 1]} = \frac{2\Pr[h(x) = y = 1]}{2\Pr[h(x) = y = 1] + \Pr[h(x) \neq y]}. \end{aligned}$$

Next, we adjust the definition of F_β to subpopulations.

Definition 2.25. *Let $h \in \{0,1\}^{\mathcal{X}}$ be a predictor, D denote be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\beta \geq 0$ be a parameter. The F_β score of predictor h and subpopulation $U \in \Gamma$ is*

$$F_\beta(h, U) = \frac{(1+\beta^2)\Pr[h(x) = y = 1, x \in U]}{\beta^2 \Pr[y = 1, x \in U] + \Pr[h(x) = 1, x \in U]}.$$

We remark that the F_1 score is the balanced $F-$score, i.e.,

$$\begin{aligned} F_1(h, U) &= \frac{2\Pr[h(x) = y = 1, x \in U]}{\Pr[y = 1 | x \in U] + \Pr[h(x) = 1, x \in U]} \\ &= \frac{2\Pr[h(x) = y = 1, x \in U]}{2\Pr[h(x) = y = 1, x \in U] + \Pr[h(x) \neq y, x \in U]}. \end{aligned}$$

Similarly, we define the empirical $F-$score of a subpopulation.

Definition 2.26. *Let $h \in \{0,1\}^{\mathcal{X}}$ be a predictor, D denote be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\beta \geq 0$ be a parameter. The empirical F_β score of predictor h and subpopulation $U \in \Gamma$ is*

$$\hat{F}_\beta(h, U, S) := (1+\beta^2)\frac{\sum_{i=1}^m \mathbb{I}[x_i \in U, h(x_i) = y_i = 1]}{\beta^2 \sum_{i=1}^m \mathbb{I}[x_i \in U, y_i = 1] + \sum_{i=1}^m \mathbb{I}[x_i \in U, h(x_i) = 1]}.$$

Next, we define Uniform Convergence for F-Score.

Definition 2.27 (F-Score Uniform Convergence)**.** *A predictor class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ has*

the F-Score uniform convergence *property (w.r.t. collection* Γ*) if there exist a function* $m^F_{\mathcal{H}}(\epsilon, \delta, \gamma, \beta) \in \mathbb{N}$*, for* $\epsilon, \delta, \gamma, \in (0, 1]$*,* $\beta \geq 0$*, such that for every distribution* D *over* $\mathcal{X} \times \{0, 1\}$*, if* $S^m = \{(x_1, y_1), \cdots, (x_m, y_m)\}$ *is a training set of* $m \geq m^F_{\mathcal{H}}(\epsilon, \delta, \gamma, \beta)$ *examples drawn i.i.d. from* D*, then for every* $h \in \mathcal{H}$ *and every subpopulation* U *s.t.* $\Pr[x \in U] \geq \gamma$*, the difference between the F-score and the empirical F-score is at most* ϵ *with probability of at least* $1 - \delta$*, i.e.,*

$$\Pr_D[|F_\beta(h, U) - \hat{F}_\beta(h, U, S)| \leq \epsilon] > 1 - \delta.$$

We finish the section with a theorem that states the sample complexity required for F-score uniform convergence for subpopulations set, Γ. The proof follows from the the same techniques we have applied in Section 2.1.3.

Theorem 2.28. *Let* $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ *be an infinite predictor class from domain* $\mathcal{X}$ *to a discrete prediction set* $\mathcal{Y}$ *with finite graph-dimension* $d_G(\mathcal{H}) \leq d$*, then* $\mathcal{H}$ *has the uniform multicalibration convergence property with* $m^F_{\mathcal{H}}(\epsilon, \delta, \gamma, \psi) = O\left(\frac{1}{\epsilon^2 \gamma}\left(d + \log\left(|\Gamma|/\delta\right)\right)\right)$.

2.1.6 Useful Definitions & Theorems

Throughout this chapter, we used the following standard Chernoff bounds.

Lemma 2.29 (Absolute Chernoff Bound)**.** *Let $X_1, ..., X_n$ be i.i.d. binary random variables with $\mathbb{E}[X_i] = \mu$ for all $i \in [n]$. Then, for any $\epsilon > 0$: $\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right] \leq 2\exp(-2\epsilon^2 n)$.*

Lemma 2.30 (Relative Chernoff Bound)**.** *Let $X_1, ..., X_n$ be i.i.d. binary random variables and let X denote their sum. Then, for any $\epsilon \in (0,1)$:$\Pr[X \leq (1-\epsilon)\mathbb{E}[X]] \leq \exp(-\epsilon^2 \mathbb{E}[X]/2)$.*

Next, the definition of Vapnik–Chervonenkis dimension, following by Uniform convergence for statistical learning and the Fundamental Theorem of Statistical Learning.

Definition 2.31. *[VC-dimension] Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. A subset $S = \{x_1, ..., x_{|S|}\} \subseteq \mathcal{X}$ is shattered by $\mathcal{H}$ if: $\left|\left\{\left(h(x_1), ..., h(x_{|S|})\right) : h \in \mathcal{H}\right\}\right| = 2^{|S|}$. The VC-dimension of $\mathcal{H}$, denoted $VCdim(\mathcal{H})$, is the maximal cardinality of a subset $S \subseteq \mathcal{X}$ shattered by $\mathcal{H}$.*

Definition 2.32 (Uniform convergence for statistical learning)**.** *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class. We say that $\mathcal{H}$ has the uniform convergence property w.r.t. loss function ℓ if there exists a function $m^{sl}_{\mathcal{H}}(\epsilon, \delta) \in \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and for every probability distribution D over $\mathcal{X} \times \{0,1\}$, if S is a sample of $m \geq m^{sl}_{\mathcal{H}}(\epsilon, \delta)$ examples drawn i.i.d. from to D, then, with probability of at least $1-\delta$, for every $h \in \mathcal{H}$, the difference between the risk and the empirical risk is at most ϵ. Namely, with probability $1-\delta$, $\forall h \in \mathcal{H} : |L_S(h) - L_D(h)| \leq \epsilon$.*

Theorem 2.33. *[The Fundamental Theorem of Statistical Learning] Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a binary hypothesis class with $VCdim(\mathcal{H}) = d$ and let the loss function, ℓ, be the*

$0-1$ *loss. Then,* $\mathcal{H}$ *has the uniform convergence property with sample complexity* $m_{\mathcal{H}}^{UC}(\epsilon, \delta) = \Theta\left(\frac{1}{\epsilon^2}\left(d + \log(1/\delta)\right)\right)$.

2.1.7 Proofs for Finite Graph Dimension Classes (Section 2.1.3)

Proof. (Proof of Lemma 2.16)
Let us assume that $VCdim(\mathcal{H}_v) > d$ and let S be a sample of size $d+1$ such that $\mathcal{H}_v$ shatters S.

Let us define the function $f : S \to \mathcal{Y}$ as:

$$\forall x \in S : f(x) = v$$

Let $T \subseteq S$ be an arbitrary subset of S. By assuming that $\mathcal{H}_v$ shatters S we know that there exists $h_v \in \mathcal{H}_v$ such that:

$$\forall x \in S : h_v(x) = 1 \iff x \in T$$

This means that for the corresponding predictor $h \in \mathcal{H}$:

$$\forall x \in S : h(x) = v = f(x) \iff x \in T$$

Thus, using our definition of f,

$$\forall T \subseteq S, \exists h \in \mathcal{H}, \forall x \in S : h(x) = f(x) \iff x \in T$$

Which means that S is G-shattered by $\mathcal{H}$. However, since $|S| > d$, it is a contradiction to the assumption that $d_G(\mathcal{H}) \leq d$. ∎

Proof. (Proof of Lemma 2.17)
Assume that $VCdim(\Phi_{\mathcal{H}_v}) > d$ and let S be a sample of $d+1$ domain points and outcomes shattered by $\Phi_{\mathcal{H}_v}$.

Note that $y = 0$ implies that $\forall h_v \in \mathcal{H}_v, \forall x \in \mathcal{X} : \phi_{h_v}(x, y) = 0$. Thus, $\forall (x, y) \in S : y = 1$ (otherwise S cannot be shattered).

Let $S_x = \{x_j : (x_j, y_j) \in S\}$. Observe that when $y = 1$, $\forall h_v \in \mathcal{H}_v, \forall x \in \mathcal{X} : \phi_{h_v}(x, 1) = h_v(x)$. Thus, the fact that S is shattered by $\Phi_{\mathcal{H}_v}$ implies that S_x is shattered by $\mathcal{H}_v$. However, $|S_x| = d + 1$. Thus, we have a contradiction to the assumption that $VCdim(\Phi_{\mathcal{H}_v}) > d$. ∎

Proof. (Proof of Lemma 2.18)
Let $\mathcal{H}_v$ and $\Phi_{\mathcal{H}_v}$ be the binary prediction and binary prediction-outcome classes of $\mathcal{H}$.

Using Lemmas 2.16 and 2.17, and since $d_G(\mathcal{H}) \leq d$, we know that $VCdim(\Phi_{\mathcal{H}_v}) \leq VCdim(\mathcal{H}_v) \leq d$.

In addition, note that:

$$\left| \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left[h(x_i) = v\right] - \Pr_{x \sim D_U}[h(x) = v] \right| = \left| \frac{1}{m} \sum_{i=1}^{m} h_v(x_i) - \Pr_{x \sim D_U}[h_v(x) = 1] \right|,$$

And

$$\left| \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left[h(x_i) = v, y = 1\right] - \Pr_{(x,y) \sim D_U}[h(x) = v, y = 1] \right| = \left| \frac{1}{m} \sum_{i=1}^{m} \phi_{h,v}(x_i, y_1) - \Pr_{(x,y) \sim D_U}[\phi_{h,v}(x, y)] \right|.$$

and the lemma follows directly from Corollary 2.13. ∎

Proof. (Proof of Lemma 2.19)
Let us denote $\xi := \psi\epsilon/3$

$$\frac{p_1}{p_2} - \frac{\tilde{p}_1}{\tilde{p}_2} \leq \frac{p_1}{p_2} - \frac{p_1 - \xi}{p_2 + \xi} = \frac{p_1(1 + \xi/p_2)}{p_2(1 + \xi/p_2)} - \frac{p_1 - \xi}{p_2(1 + \xi/p_2)} = \frac{\xi}{p_2(1 + \xi/p_2)} \left[\frac{p_1}{p_2} + 1\right]$$

Since $p_1, \psi \leq p_2$,

$$\frac{\xi}{p_2(1 + \xi/p_2)} \left[\frac{p_1}{p_2} + 1\right] \leq \frac{\xi}{p_2} \left[\frac{p_2}{\psi} + \frac{p_2}{\psi}\right] = \frac{2\xi}{\psi} \leq \frac{3\xi}{\psi} = \epsilon.$$

Similarly,

$$\frac{\tilde{p}_1}{\tilde{p}_2} - \frac{p_1}{p_2} \leq \frac{p_1 + \xi}{p_2 - \xi} - \frac{p_1}{p_2} = \frac{p_1 + \xi}{p_2(1 - \xi/p_2)} - \frac{p_1(1 - \xi/p_2)}{p_2(1 - \xi/p_2)} = \frac{\xi}{p_2(1 - \xi/p_2)} \left[1 + \frac{p_1}{p_2}\right].$$

Since $p_1, \psi \leq p_2$,

$$\frac{\xi}{p_2(1 - \xi/p_2)} \left[1 + \frac{p_1}{p_2}\right] \leq \frac{\xi}{p_2(1 - \xi/\psi)} \left[\frac{p_2}{\psi} + \frac{p_2}{\psi}\right] = \frac{2\xi}{\psi(1 - \xi/\psi)} = \frac{2\epsilon}{3(1 - \epsilon/3)} \leq \frac{2\epsilon}{3(1 - 1/3)} = \epsilon$$

Thus,

$$\left|\frac{p_1}{p_2} - \frac{\tilde{p}_1}{\tilde{p}_2}\right| \leq \epsilon$$

■

Proof. (Proof of Lemma 2.20) Let P_U denote the probability of subpopulation U:

$$\mathrm{P}_U := \Pr_{x \sim D}[x \in U]$$

Using the relative Chernoff bound (Lemma 2.30) and since $\mathbb{E}[|S \cap U|] = m\mathrm{P}_U$, we can bound the probability of having a small sample size in U. Namely, if $\mathrm{P}_U \geq \gamma$, then:

$$\Pr_D\left[|S \cap U| \leq \frac{\gamma m}{2}\right] \leq \Pr_D\left[|S \cap U| \leq \frac{m\mathrm{P}_U}{2}\right] \leq e^{-\frac{m\mathrm{P}_U}{8}} \leq e^{-\frac{\gamma m}{8}}$$

Thus, for any $U \in \Gamma_\gamma$, if $m \geq \frac{8\log\left(\frac{|\Gamma|}{\delta}\right)}{\gamma}$, then, with probability of at least $1 - \frac{\delta}{|\Gamma|}$,

$$|S \cap U| > \frac{\gamma m}{2}$$

Finally, using the union bound, with probability at least $1 - \delta$, for all $U \in \Gamma_\gamma$,

$$|S \cap U| > \frac{\gamma m}{2}$$

■

Proof. (Proof of Theorem 2.11)

Let $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ be a sample of m labeled examples drawn i.i.d. according to D, and let $S_U := \{(x, y) \in S : x \in U\}$ be the samples in S that belong to subpopulation U.

Let Γ_γ denote the set of all subpopulations $U \in \Gamma$ that has probability of at least γ:

$$\Gamma_\gamma := \{U \in \Gamma \mid \Pr_{x \sim D}[x \in U] \geq \gamma\}$$

Let us assume the following lower bound on the sample size:

$$m \geq \frac{8\log\left(\frac{2|\Gamma|}{\delta}\right)}{\gamma}$$

Thus, using Lemma 2.20, we can bound the probability of having a subpopulation $U \in \Gamma_\gamma$ with small number of samples. Namely, we know that with probability of at least $1 - \delta/2$, for every $U \in \Gamma_\gamma$:

$$|S_U| \geq \frac{\gamma m}{2}$$

Next, we would like to show that having a large sample size in U implies accurate approximation of the calibration error, with high probability, for any interesting category in (U, I). For this purpose, let us define ϵ', δ' as:

$$\epsilon' := \frac{\psi\epsilon}{3}$$

$$\delta' := \frac{\delta}{4|\Gamma||\mathcal{Y}|}$$

By using Lemma 2.18 and since $d_G(\mathcal{H}) \leq d$, we know that there exists some constant $a > 0$, such that, for any $v \in \mathcal{Y}$ and any $U \in \Gamma_\gamma$, with probability at least $1 - \delta'$, a random sample of m_1 examples from U, where,

$$m_1 \geq a\frac{d + \log(1/\delta')}{\epsilon'^2} = 9a\frac{d + \log(\frac{4|\Gamma||\mathcal{Y}|}{\delta})}{\epsilon^2\psi^2}$$

will have,

$$\forall h \in \mathcal{H} : \left|\frac{1}{m_1}\sum_{x' \in S_U} \mathbb{I}\left[h(x') = v\right] - \Pr\left[h(x) = v \mid x \in U\right]\right| \leq \epsilon' = \frac{\psi\epsilon}{3}$$

By using Lemma 2.18 and since $d_G(\mathcal{H}) \leq d$, we know that for any $v \in \mathcal{Y}$ and any $U \in \Gamma_\gamma$, with probability at least $1 - \delta'$, a random sample of m_2 labeled examples from $U \times \{0, 1\}$, where,

$$m_2 \geq a\frac{d + \log(1/\delta')}{\epsilon'^2} = 9a\frac{d + \log(\frac{4|\Gamma||\mathcal{Y}|}{\delta})}{\epsilon^2\psi^2}$$

will have,

$$\forall h \in \mathcal{H} : \left| \frac{1}{m_2} \sum_{(x',y') \in S_U} \mathbb{I}\left[h(x') = v, y' = 1\right] - \Pr\left[h(x) = v, y = 1 \mid x \in U\right] \right| \leq \epsilon' = \frac{\psi\epsilon}{3}$$

Let us define the constant a' in a manner that sets an upper bound on both m_1 and m_2:

$$a' := 18a$$

and let m' be that upper bound:

$$m' := a' \frac{d + \log\left(\frac{|\Gamma||\mathcal{Y}|}{\delta}\right)}{\psi^2\epsilon^2} \geq \max(m_1, m_2)$$

Then, by the union bound, if for all subpopulation $U \in \Gamma_\gamma$, $|S_U| \geq m'$, then, with probability at least $1 - 2|\Gamma||\mathcal{Y}|\delta' = 1 - \frac{\delta}{2}$:

$$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y} :$$
$$\left| \frac{1}{|S_U|} \sum_{(x',y') \in S_U} \mathbb{I}\left[h(x') = v\right] - \Pr\left[h(x) = v \mid x \in U\right] \right| \leq \frac{\psi\epsilon}{3}$$
$$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y} :$$
$$\left| \frac{1}{|S_U|} \sum_{(x',y') \in S_U} \mathbb{I}\left[h(x') = v, y' = 1\right] - \Pr\left[h(x) = v, y = 1 \mid x \in U\right] \right| \leq \frac{\psi\epsilon}{3}$$

Let us choose the sample size m as follows:

$$m := \frac{2m'}{\gamma} = 2a \frac{d + \log\left(\frac{|\Gamma||\mathcal{Y}|}{\delta}\right)}{\psi^2\epsilon^2\gamma}$$

Recall that with probability at least $1 - \delta/2$, for every $U \in \Gamma_\gamma$:

$$|S_U| \geq \frac{\gamma m}{2} = m'$$

Thus, using the union bound once again, with probability at least $1 - \delta$:

$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y}$:

$$\left| \frac{1}{|S_U|} \sum_{x' \in S_U} \mathbb{I}\left[h(x') = v\right] - \Pr\left[h(x) = v \mid x \in U\right] \right| \leq \frac{\psi\epsilon}{3}$$

$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y}$:

$$\left| \frac{1}{|S_U|} \sum_{(x',y') \in S_U} \mathbb{I}\left[h(x') = v, y' = 1\right] - \Pr\left[h(x) = v, y = 1 \mid x \in U\right] \right| \leq \frac{\psi\epsilon}{3}$$

To conclude the theorem, we need show that having $\psi\epsilon/3$ approximation to the terms described above, implies accurate approximation to the calibration error. For this purpose, let us denote:

$$\begin{aligned}
p_1(h, U, v) &:= \Pr\left[h(x) = v, y = 1 \mid x \in U\right] \\
p_2(h, U, v) &:= \Pr\left[h(x) = v \mid x \in U\right] \\
\tilde{p}_1(h, U, v) &:= \frac{1}{|S_U|} \sum_{(x',y') \in S_U} \mathbb{I}\left[h(x') = v, y' = 1\right] \\
\tilde{p}_2(h, U, v) &:= \frac{1}{|S_U|} \sum_{x' \in S_U} \mathbb{I}\left[h(x') = v\right]
\end{aligned}$$

Then, with probability at least $1-\delta$:

$$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y} : \left| \tilde{p}_2(h,U,v) - p_2(h,U,v) \right| \leq \frac{\psi\epsilon}{3}$$
$$\forall h \in \mathcal{H}, \forall U \in \Gamma_\gamma, \forall v \in \mathcal{Y} : \left| \tilde{p}_1(h,U,v) - p_1(h,U,v) \right| \leq \frac{\psi\epsilon}{3}$$

Using Lemma 2.19, for all $h \in \mathcal{H}$, $U \in \Gamma_\gamma$ and $v \in \mathcal{Y}$, if $p_2(h,U,v) \geq \psi$, then:

$$\left| \frac{p_1(h,U,v)}{p_2(h,U,v)} - \frac{\tilde{p}_1(h,U,v)}{\tilde{p}_2(h,U,v)} \right| \leq \epsilon$$

Thus, since

$$c(h,U,\{v\}) = \frac{p_1(h,U,v)}{p_2(h,U,v)} - v$$
$$\hat{c}(h,U,\{v\},S) = \frac{\tilde{p}_1(h,U,v)}{\tilde{p}_2(h,U,v)} - v$$

then with probability at least $1-\delta$:

$$\forall h \in \mathcal{H}, \forall U \in \Gamma, \forall v \in \mathcal{Y} :$$
$$\Pr[x \in U] \geq \gamma, \Pr\left[h(x) = v \mid x \in U\right] \geq \psi \Rightarrow |c(h,U,\{v\}) - \hat{c}(h,U,\{v\},S)| \leq \epsilon$$

■

2.1.8 Proofs for Lower Bounds (Section 2.1.4)

Proof. (Proof of Theorem 2.12) Let $\mathcal{X} = \{x^0, x^1, x^2\}$, let $U = \{x^0, x^1\}$ and let $H = \{h\}$, where

$$h(x) = \begin{cases} \frac{1}{2} + \epsilon & x = x^0 \\ 0 & else. \end{cases}$$

Let $\Gamma = \{U, \{x^2\}\}$. Let $D \in \{D_1, D_2\}$ where

$$D_1(x,y) = \begin{cases} (1/2+\epsilon)\psi\gamma & (x,y) = (x^0, 1) \\ (1/2-\epsilon)\psi\gamma & (x,y) = (x^0, 0) \\ (1-\psi)\gamma & (x,y) = (x^1, 0) \\ 1-\gamma & (x,y) = (x^2, 0) \end{cases}$$

and

$$D_2(x,y) = \begin{cases} (1/2+\epsilon)\psi\gamma & (x,y) = (x^0, 0) \\ (1/2-\epsilon)\psi\gamma & (x,y) = (x^0, 1) \\ (1-\psi)\gamma & (x,y) = (x^1, 0) \\ 1-\gamma & (x,y) = (x^2, 0) \end{cases}$$

Now we will show a reduction to coin tossing:
Consider two biased coins. The first coin has a probability of $r_1 = 1/2 + \epsilon$ for heads and the second has a probability of $r_2 = 1/2 - \epsilon$ for heads. We know that in order to distinguish between the two with confidence $\geq 1 - \delta_1$, we need at least $C\frac{\ln(\frac{1}{\delta_1})}{\epsilon^2}$ samples.

Since

$$\Pr_{(x,y)\sim D}[x \in U] = \Pr_{(x,y)\sim D}[x \neq x^2] = \gamma$$

the first condition for multicalibration holds. Now, we use another property of our "tailor-maded" distribution D and single predictor h, which is $\{x \in \mathcal{X} : h(x) = \frac{1}{2} + \epsilon\} = \{x \in \mathcal{X} : h(x) = \frac{1}{2} + \epsilon, x \in U\} = \{x_0\}$, to get the second condition:

$$\Pr_D[h(x) = 1/2 + \epsilon | x \in U] = \Pr_D[x = x^0 | x \in U] = \frac{\psi\gamma}{\gamma} = \psi,$$

and that

$$\Pr_D[y = 1 | h(x) = \frac{1}{2} + \epsilon, x \in U] = \Pr_D[y = 1 | x = x^0]$$

is either $1/2 + \epsilon$ (if $D = D_1$) or $1/2 - \epsilon$ (in case $D = D_2$) (recall that $D \in \{D_1, D_2\}$).

Now, if H has the multicalibration uniform convergence property with a sample $S = (x_i, y_i)_{i=1}^m$ of size m, and if

$$\sum_{i=1}^{m} \frac{\mathbb{I}[y_i = 1, h(x_i) = 1/2 + \epsilon, x_i \in U]}{\sum_{j=1}^{m} \mathbb{I}[h(x_i) = 1/2 + \epsilon, x_i \in U]} = \sum_{i=1}^{m} \frac{\mathbb{I}[y_i = 1, x_i = x^0]}{\sum_{j=1}^{m} \mathbb{I}[x_i = x^0]} > \frac{1}{2}$$

holds, then

$$\Pr[y = 1 | h(x) = \frac{1}{2} + \epsilon, x \in U] = \frac{1}{2} + \epsilon$$

holds w.p. $1 - \delta_1$ (from the definition of multicalibration uniform convergence).

Let us assume by contradiction that we can get multicalibration uniform convergence with $m = \frac{C}{\epsilon^2\psi\gamma} - \frac{k}{\psi\gamma} < \frac{C}{\epsilon^2\psi\gamma}$ for some constant $k = \Omega(1)$.

Let m_0 denote the random variable that represents the number of samples in S such that $x_i = x^0$ (i.e., $h(x_i) = 1/2 + \epsilon$). Hence, $\mathbb{E}[m^0] = \gamma \cdot \psi \cdot m = \frac{C}{\epsilon^2} - k$.

From Hoeffding's inequality,

$$\Pr[m^0 \geq \frac{C}{\epsilon^2}] = \Pr[m^0 - \underbrace{(\frac{C}{\epsilon^2} - k)}_{\mathbb{E}[m_0]} \geq k] \leq e^{-2mk^2}.$$

Let δ_2 be the parameter that holds $e^{-2mk^2} \leq \delta_2$, and let $\delta := \delta_1 + \delta_2$. Then we get that with probability $> (1-\delta_1)(1-\delta_2) > 1 - \delta_1 - \delta_2 = 1 - \delta$ we can distinguish between the two coins with less than $\frac{C}{\epsilon^2}$ samples, which is a contradiction.

■

Proof. (Proof of Theorem 2.22)

We will show that for every $\epsilon < 3/8$ and every $\delta < 1$ we have that $m(\epsilon, \delta) > d/8$.
Let $\mathcal{Y} = \{0, 1\}$ and $\Gamma = \{U\} = \{\mathcal{X}\} = \{\{x^1, \ldots, x^d\}\}$. Notice that since $d_N(\mathcal{H}) = d$ and $\mathcal{X} = U$ is also of size d, it must be that U is a set which is $N-$shattered set by $\mathcal{H}$.
Let D be a distribution such that for every $j \in [d]$,

$$D(x, y) = \begin{cases} \frac{1}{2d} & (x, y) = (x^j, 1) \\ \frac{1}{2d} & (x, y) = (x^j, 0) \end{cases}$$

Let $S = (x_i, y_i)_{i=1}^m$ be a sequence of $m = \frac{d}{8}$ i.i.d. samples from D.
Notice that since U is $N-$shattered by $\mathcal{H}$ there exists $h_S \in \mathcal{H}$, such that

$$h_S(x) = \begin{cases} 1 & \exists i : x_i = x \text{ and } (x_i, 1) \in S \\ 0 & else. \end{cases}$$

Since

$$\Pr_D[x \in U] = 1 \geq \frac{1}{2} \geq \gamma,$$

and

$$\Pr_D[h_S(x) = 0 | x \in U] = \Pr_D[h_S(x) = 0] \geq \Pr_D[(x, 1) \notin S \wedge (x, 0) \notin S] \geq \frac{7}{8} \geq \psi.$$

So $(U, \{0\})$ is an interesting category.

We move on the calculate the multicalibration error.

First,

$$c(h_S, U, \{0\}) = \frac{1}{2} \geq \frac{3}{8}.$$

Now, from the way we selected h_S we have that for every $i \in [d/8]$ we have that $y_i = 1$ yields $h(x_i) = 0$, which means that $\mathbb{I}[h(x_i) = 0]y_i = 0$ for every $i \in [d/8]$. Hence,

$$\hat{c}(h_S, U, \{0\}) = 0$$

Put it all together, we have that with probability 1,

$$|\hat{c}(h_S, U, \{0\}) - c(h_S, U, \{0\})| \geq \frac{3}{8}.$$

Hence the sample size for uniform convergence is at least $d/8$. ∎

Proof. (Proof of Theorem 2.23) Assume $\mathcal{X} = \{x_1, \ldots, x_d\}$, $\mathcal{Y} = \{0, 1\}$, $\mathcal{H} = \{h_0, h_1\}$, where h_y is the predictor that maps all $x \in \mathcal{X}$ to y, i.e., $h_y(x) = y$. Assume $\Gamma = 2^{\mathcal{X}}$, i.e. the collection of subpopulations is the power set of $\mathcal{X}$. Notice that $\log |\Gamma| = d$. In addition, assume a distribution D such that $\forall x \in \mathcal{X}$,

$$\Pr_{(x,y)\sim D}[x_i = x, y_i = 1] = \Pr_{(x,y)\sim D}[x_i = x, y_i = 0] = \frac{1}{2d}.$$

Let $S = (x_i, y_i)_{i=1}^m$ be a sequence of $m = \frac{d}{2}$ i.i.d. samples from D.

Denote

$$U_S = \mathcal{X} \setminus S.$$

We have that $U_S \in \Gamma$, and since $|U_S| \geq \frac{d}{2}$,

$$\Pr_D[x \in U_S] \geq \frac{1}{2} \geq \gamma.$$

In addition,

$$\Pr_D[h_0(x) = 0 | x \in U_S] = \Pr_D[h_1(x) = 1 | x \in U_S] = 1 \geq \psi,$$

therefore both $(U_S, \{0\}), (U_S, \{1\})$ are interesting categories.

However,

$$\hat{c}_S(h_0, U_S, \{0\}) = 0,$$

and

$$c(h_0, U_S, \{0\}) = \frac{1}{2}.$$

Hence, w.p. 1,

$$|\hat{c}(h_S, U_S, \{0\}) - c(h_S, U_S, \{0\})| \geq \frac{1}{2}.$$

As a result, the sample size for uniform convergence is at least $\Omega(d) = \Omega(\log(|\Gamma|)$. ■

Chapter 3

Online Learning

Introduction

Algorithms for online learning are especially challenging due to various constraints the learner needs to comply with: bounded computation time, handling an unlimited amount of data, optimizing the performance throughout the entire learning phase (regret minimization), and up to the ability to adapt to dynamic environments that depend on the actions of the learner (Reinforcement Learning).

Reinforcement Learning (RL) is concerned with online learning where the goal of the learner is to map signals of the current state into actions in a way that maximizes the cumulative reward. The goal of the learner in RL setting is either to minimize sample complexity to find an optimal policy or to minimize the regret, where regret is the difference between the online cumulative reward and that of the best action. Common settings for RL problems include the Markov Decision Process (MDP) model, the Multi-Arm Bandits (MAB) model, and the Dueling Bandits model (which I briefly describe before the relevant discussion on the matter). Next, I will give a brief overview of my work in online learning.

Dueling Bandits model: In Dueling bandits [133], the realization of the rewards of the selected arm is no longer the feedback as in the case of MAB. As an alternative, the learner chooses a pair of arms and observes the winning arm of their "duel". We assume that for each pair of arms $i, j \in [n]$ there exists an (unknown) distinguishability $\Delta_{i,j} \in [-1/2, 1/2]$ such that the probability that arm i wins in a duel against arm j is $1/2 + \Delta_{i,j}$, and that $\Delta_{j,i} = 1 - \Delta_{i,j}$. Other common assumptions for this setting are (1) total order over the arms, and (2) *Strong Stochastic Transitivity* (SST) and *Strong Triangle inequality* (STI), both defined w.r.t. this total order. Together they imply that for every triplet of arms $a \succ b \succ c$ it holds that $\max\{\Delta_{a,b}, \Delta_{b,c}\} \leq_{(SST)} \Delta_{a,c} \leq_{(STI)} \Delta_{a,b} + \Delta_{b,c}$.

Dueling Bandits with Team Comparisons In a paper that was published in NeurIPS 21 [42] we introduced the *dueling teams problem*, a new online-learning setting in which the learner observes noisy comparisons of disjoint pairs of k-sized *teams* (subsets of arms) from a universe of n players (arms). The goal of the learner is to minimize the number of duels required to identify, with high probability, a *Condorcet winning team*, i.e., a team that wins against any other disjoint team (with probability at least 1/2). This appears naturally in sports or online games, where the goal is to pick one of the best teams from a set of players by observing the outcomes of matches. We formalize the model and provide several algorithms, both for stochastic and deterministic settings. For the stochastic setting, we provided a reduction to the classical dueling bandits setting, yielding an efficient algorithm that identifies a Condorcet winning team within $\mathcal{O}((n + k\log(k))\frac{\max(\log\log n, \log k)}{\Delta^2})$ duels, where Δ is a gap parameter. Building on this, we also derive an upper bound for the regret of the problem. For deterministic feedback, we present a gap-independent algorithm that identifies a Condorcet winning team within $\mathcal{O}(nk\log(k) + k^5)$ duels.

Generalizing Dueling Bandits framework (future work) We plan to use the insights gained by our above preliminary work and derive optimal regret and sample complexity bounds to identify the best arm for more general settings than [133]. In particular, we plan to consider total order with either SST or STI but not both.

Modeling Attrition in Recommender Systems with Departing Bandits Traditionally, when recommender systems are formalized as multi-armed bandits, the policy of the recommender system influences the rewards accrued, but not the length of interaction. However, in real-world systems, dissatisfied users may depart (and never come back). In a work that appeared in AAAI' 22 [18], we proposed a novel multi-armed bandit setup that captures such policy-dependent horizons. Our setup consists of a finite set of user *types*, and multiple arms with Bernoulli payoffs. Each (user type, arm) tuple corresponds to an (unknown) reward probability. Each user's type is initially unknown and can only be inferred through their response to recommendations. Moreover, if a user is dissatisfied with their recommendation, they might depart the system. We first address the case where all users share the same type, demonstrating that a recent UCB-based algorithm is optimal. We then move forward to the more challenging case, where users are divided among two types. While naive approaches cannot handle this setting, we provide an efficient learning algorithm that achieves $\tilde{O}(\sqrt{T})$ regret, where T is the number of users.
This problem demonstrates nicely how simply taking into account the less popular preferences of users can lead to significant improvement in the performance of the learner.

Finding Safe Zones of Markov Decision Processes Policies [1] Given a policy, we define a SafeZone as a subset of states, such that most of the policy's trajectories

[1] Joint work with Michal Moshkovitz and Yishay Mansour, in submission.

are confined to this subset. The quality of the SafeZone is parameterized by the number of states and the escape probability, i.e., the probability that a random trajectory will leave the subset. SafeZones are especially interesting when they have a small number of states and low escape probability. We study the complexity of finding optimal SafeZones, and show that in general, the problem is computationally hard. For this reason, we concentrate on computing approximate SafeZones. Our main result is a bi-criteria approximation algorithm which gives a factor of almost 2 approximation for both the escape probability and SafeZone size, using a polynomial size sample complexity. We concluded this work with an empirical evaluation of our algorithm.

The problem we introduced addresses anomaly detection and safe RL, and can also be viewed through the lens of explainable RL.

Finding policies with small SafeZones (future work) An interesting direction for future work I plan to peruse is the following. Given an upper bound over the escape probability $\rho > 0$ and an MDP (known or unknown to the learner), find a policy for the MDP with a small SafeZone and an escape probability bounded from above by ρ. An interesting observation that came up from our empirical demonstration is that different policies result in different sizes of SafeZones, and that the optimal policy does not necessarily has the smallest SafeZone.

Learning the best team, attrition in recommender systems, and safety in reinforcement learning are all related to optimization in ML systems that revolves around people. In learning the best team, the objective is to determine the most effective combination of team members or agents to work together to achieve a common goal. This requires optimizing the performance of individuals in a special manner as it is impossible to test the performance of a team of individuals against the same team with only a

single member different.

In recommender systems, attrition refers to the loss of customers or users over time. The goal is to optimize personalized recommendations to retain users and improve customer satisfaction. This involves identifying user preferences (that might contribute to attrition) and developing strategies to mitigate them.

Finally, in reinforcement learning, safety is a critical consideration when training agents to make decisions in complex and dynamic environments. In the anomaly detection approach we take, we prioritize the popular states of the Markov model, thus avoiding states that could be unsafe due to their unpopularity.

Overall, these works are all related to the optimization of systems in ML with societal solutions. They highlight the importance of balancing individual constraints and system objectives to achieve optimal outcomes. Additionally, they all involve the consideration of interactions between individuals and the system, as well as the impact of these interactions on the overall system's performance.

3.1 Dueling Teams

3.1.1 Introduction

Multi-arm bandits (MAB) is a classical model of decision making under uncertainty. In spite of the simplicity of the model, it already incorporates the essential tradeoff between exploration and exploitation. In MAB, the learner performs actions and can only observe rewards of the actions performed. One of the main tasks in MAB is *best arm identification*, where the goal is to identify a near-optimal action while minimizing the number of actions executed. The MAB model has numerous practical applications, including online advertising, recommendation systems, clinical trials,

and more. (See [120, 89] for more background).

One weakness of the MAB model is the assumption that real-valued rewards are always available. In many applications, it is more natural to compare two actions and observe which one of them is better rather than give every single action a numerical reward. For example, recommendation systems often suggest two items and obtain only their relative preference as feedback (e.g., by a click on one of them). This leads very naturally to the well-known model of dueling bandits [134], where the learner selects a pair of actions each time and observes the binary "winner" of a duel between the two. See [28] for a survey on extensions of this model.

In this work we were interested in the case that the learner has to select two disjoint *teams* for a duel, which are k-sized subsets of the actions (which we call players). This appears naturally in sports or online games, where the goal is to pick one of the best teams from a set of players by observing the outcomes of matches (say, to be a school representative team, or to sponsor for tournaments). Examples include doubles tennis, basketball, and the online game League of Legends, where each match requires two disjoint teams of players to compete. Similar phenomena appear in working environments, where different R&D teams compete on implementing a project. Another example could be online advertisements where multiple products are bundled to a display ad and a customer can click on any of two presented bundles, e.g., some online games offer in-app bundle purchases, and the information regarding sales of different bundles can be used to improve the bundles' composition.

Our basic model is the following. We have a universe of n players, and at each iteration the learner selects two **disjoint** teams for a duel and observes the winner. For any two different teams, there exists an unknown stationary probability that determines the winner of a duel between them. The requirement that teams need to be disjoint is in accordance with the situation in games, where a single person cannot

play for both teams. The goal of the learner is to minimize the number of duels required to identify, with high probability, a *Condorcet winning team*, i.e., a team which wins against any other disjoint team (with a probability of at least 1/2). We do assume these probabilities are linked to a strict total order on all teams. This implies the existence of a Condorcet winning team, yet it is typically not unique. We make two minimal and natural assumptions on this total order on teams, namely, that it is *consistent* to some total order among the players, and that the team probabilistic comparisons hold *Strong Stochastic Transitivity*, a common assumption in dueling bandit settings.

Clearly, given any total order among the players, the best team is the one containing the top k players, which is in particular one of the Condorcet winning teams. However, not all relations between players are deducible for the learner. In particular, even achieving accurate estimations of the latent winning probabilities between all disjoint teams might not suffice to separate the top k players from the rest. Consider for example an instance with four players $1 \succ 2 \succ 3 \succ 4$ where $k = 2$ and the total order among the teams is lexicographical, i.e., $12 \succ 13 \succ 14 \succ 23 \succ 24 \succ 34$. Then, there exist three feasible duels, each of which is won by the team containing player 1 with probability greater than 1/2. If all three duels are won with equal probability by the team containing 1, the learner has no chance of detecting the team 12 as the top k team. However, any of the teams $12, 13$ and 14 is a Condorcet winning team.

Our main target is to present algorithms for which the number of duels is bounded by a polynomial in the number of players n and team size k, although the number of teams is exponential in k, i.e., $\Omega((\frac{n}{k})^k)$ and the number of valid duels is $\Omega(2^k(\frac{n}{2k})^{2k})$. Even if one were to accept an exponential number of arms, a direct reduction to the standard dueling bandits setting would not be feasible as not all pairs of teams are comparable in our model. In particular, duels of the form $(S \cup \{a\}, S \cup \{b\})$, which

would yield a signal regarding the relation between players a and b, are forbidden. The inherent difficulty of our endeavor comes from two limitations: (1) Not all the relations between two single players are deducible, (see example above), and (2) even for pairs of players with deducible relation, having $\Omega(2^k(\frac{n}{2k})^{2k})$ valid duels and the same amount of (latent) winning probabilities makes the task of deducing their relations hard.

We start by giving a full characterization of the *deducible* pairwise relations between players, namely relations that can be detected by a learner which is allowed to perform an unlimited amount of duels. Our characterization implies that every deducible single player relation has one of two types of *witnesses*, which are constant-size sets of duels that prove their relation. We also show that, once we find a witness for one pair of players, it can often be transferred to a witness for other pairs of players.

Building upon this characterization, we introduce a parameter $\Delta_{a,b}$ which captures the distinguishability of any two players a and b and takes a value of 0 whenever the pair is not deducible. Assuming $\Delta := \Delta_{k,k+1} > 0$, where k and $k+1$ are k^{th} and $(k+1)^{\text{th}}$ best players, we give a reduction to the classic dueling bandits problem. Combining this reduction with a high-probability top-k identification algorithm for the dueling bandits setting (e.g., [100, 112]) yields a similar sample complexity upper bound, e.g., this yields a high-probability top-k identification algorithm for dueling teams with $\mathcal{O}(\Delta^{-2}(n + k\log(k))\max(\log\log n, \log k))$ duels.

Interestingly, it turns out that the deterministic case, i.e., when winning probabilities are in $\{0, 1\}$, constitutes a challenging special case of our problem where Δ can be particularly small, or even 0. To overcome this issue we design delicate algorithms which are independent of Δ. On a high level, a preprocessing procedure first excludes as many bad players as possible. To do so, it runs a method for identifying pairwise relations between players which performs only a small number of duels, but has little

control over the pair for which the relation is uncovered. For general total orders this implies an algorithm requiring $\mathcal{O}(nk\log(k) + 2^{\mathcal{O}(k)})$ duels. For the natural case of *additive linear* orders, we present a more elaborated approach for detecting a Condorcet winning team within the reduced instance, resulting in an algorithm that performs $\mathcal{O}(nk\log(k) + k^5)$ duels.

We introduce our problem in Section 3.1.2, give a characterization of deducible relations in Section 3.1.3, discuss the stochastic setting in Section 3.1.4, and the deterministic setting in Section 3.1.5. For brevity, algorithms and (full) proofs are relegated to Sections 3.1.6, 3.1.7, and 3.1.8. Section 5.2.1 contains a discussion and Section 3.1.8 a characterization of additive linear total orders.

Related Work

MAB best arm or subset identification: single arm identification was initiated in [52] and later studied in many works including [26, 78, 33]. This setting was extended by [75] for multiple arms identification (i.e., top k arms), using a single arm samples. Other works that address the objective of top$-k$ identification include [34, 137, 27].

Dueling bandits The work of [134] lay down the framework of non-parametric bandit feedback under total order among arms, strong stochastic transitivity, and stochastic triangle inequality assumptions and were followed by many subsequent works (For more, see a survey, [28].) In particular, some subsequent works target the task of identifying the top k players in this setting [100, 112].

Dueling bandits with sets of actions One line of dueling bandits extension consider the case where the learner selects a subset of actions and observes the outcomes of all duels between all pairs of actions in the subset [24, 122], or the winner of the subset [115, 113]. As a consequence, these settings give the learner strictly more

information than the dueling bandits setting. In contrast, feedback in our setting reveals less information.

MAB with multiple actions selection : There are works in which the learner selects a (sometimes fixed-sized) subset of actions at each iteration, and observes either all of the individual selected arms rewards (semi-bandit feedback) or an aggregated form of the rewards (full-bandit feedback), and the task is to detect to best arm or the top k. These include *combinatorial bandits* [30], *top-k* [111], *linear bandit and routing* [7], and more. The main difference between combinatorial bandits and our setting is the feedback.

Comparison models: Noisy pairwise comparison models, especially for sorting and ranking, have a long history which dates backs to the 1950's (For more, see a survey, [105].). Specifically, the mathematical problem Counterfeit coin was introduced in the form of a puzzle [61]: given a pile of 12 coins, determine which coins has a different weight (and therefore counterfeit) using balance scales while minimizing the number of measurements. The problem was followed by numerous generalizations (see [62]). While this problem is restricted to coins with two different weights, our setting can be seen as a variant with multiple weights.

3.1.2 Dueling Teams: Problem Formulation

We formalize our problem as follows. Let $n, k \in \mathbb{N}$ with $1 \leq k \leq \frac{n}{2}$. We denote the set of players by $[n] := \{1, \ldots, n\}$ and call any set of k distinct players a *team*. Moreover, we assume the existence of an underlying strict total order among all teams, and denote it by $\succ$. We also refer to $\succ$ as the ground truth order. In particular, for any two teams A and B either $A \succ B$ holds, in which case we say that A is *better* than B, or vice versa, and this relation is transitive. Additionally, we require the total order among the teams to be consistent with a total order among players and formalize

this in the *consistency* assumption at the end of this section.

In each round, the learner selects an ordered pair of two disjoint teams, A and B to perform a *duel*, and receives a noisy binary feedback about which team is better. Note that in contrast to the usual dueling bandits setting, our setting does not allow duels of the form (A, A), as selecting teams with mutual players for a duel is not an option. We denote the *observable* part of $\succ$ by $\succ_{obs}$, i.e., $A \succ_{obs} B$ iff A and B are disjoint teams and $A \succ B$. Note, $\succ_{obs}$ is not transitive, thus not even a partial order.

The outcome of a duel (A, B) is sampled independently from a Bernoulli random variable with success probability $P_{A,B}$, so in particular it holds that $P_{B,A} = 1 - P_{A,B}$. We assume that the probabilistic comparisons are linked to the total order among the teams, i.e., $A \succ B$ implies $P_{A,B} > 1/2$, and that $P_{A,B}$ exists for every pair of teams (not only disjoint ones). In the stochastic setting, we denote $A > B$ if team A is the random winner of duel (A, B). In the deterministic setting, it holds that $P_{A,B} \in \{0, 1\}$ for any teams $A \neq B$. In other words, for two disjoint teams A and B the learner can observe whether $A \succ_{obs} B$ or $B \succ_{obs} A$ by performing a single duel.

A team A is a *Condorcet winning team*[2] if $A \succ_{obs} B$ for all teams B such that $A \cap B \neq \emptyset$. From our assumption on $\succ$, there always exists a Condorcet winning team, but it is not necessarily unique. The learner's goal is to minimize the number of duels required to identify, with high probability in the stochastic setting and with probability 1 in the deterministic case, a Condorcet winning team.

In the following we formalize two more assumptions we impose on our model, the former affects the linking of the probabilities to the strict total order $\succ$, the latter

[2]The name is motivated by the fact that such a team is a weak Condorcet winner for the relation $\succ_{obs}$.

restricts the total order $\succ$ itself.

Strong stochastic transitivity (SST): Similarly to the dueling bandits settings in [134], we assume *strong stochastic transitivity*. Namely, for every triplet of different teams $A \succ B \succ C$ it holds that $P_{A,C} \geq \max\{P_{A,B}, P_{B,C}\}$.

Consistency: We assume that the total order $\succ$ is consistent to a total order among single players. More precisely, we say that $\succ$ satisfies *consistency* if for every two players $a, b \in [n]$ either of the following holds true:

(i) $S \cup \{a\} \succ S \cup \{b\}$ for all $S \subseteq [n] \setminus \{a, b\}, |S| = k - 1$.

(ii) $S \cup \{b\} \succ S \cup \{a\}$ for all $S \subseteq [n] \setminus \{a, b\}, |S| = k - 1$.

The consistency assumption lets us derive a relation among the single players, by defining $a \succ b$ iff $S \cup \{a\} \succ S \cup \{b\}$ holds for some S. By team relation transitivity, $\succ$ implies a total order on $[n]$. Whenever we write $a \succ b$ for some players $a, b \in [n]$ this is short-hand notation for $S \cup \{a\} \succ S \cup \{b\}$ for all subsets $S \subseteq [n] \setminus \{a, b\}$ of size $k - 1$. For notational convenience, we assume without loss of generality that $1 \succ 2 \succ \cdots \succ n$ and write A^*_m for the set of players containing the top m players, i.e., $A^*_m = [m]$. In particular, the consistency assumption yields that A^*_k is a Condorcet winning team.

Though the ground truth ranking induces a total order among the players, the learner might not be able to deduce the entire order. In the following we give a characterization of the *deducible* part of $\succ$.

3.1.3 Witnesses: A Characterization of Deducible Relations

In this section we provide a high level description of the complete characterization of all the pairwise relations between single players that can be deduced via team

duels. Though single players cannot be observed via team duels directly, we show a sufficient and necessary condition for deducible relations in the form of a constant number of winning probabilities of observable (feasible) duels. We refer to a set of players participating in such duels as *witnesses*. For completeness, we point out that a similar characterization can be done for any same-sized subsets of size less than k.

We denote by $\mathcal{C}_{obs}$ the set of strict total orders which are compatible with the winning probabilities of observable duels, i.e., $\{P_{A,B} \mid A \text{ and } B \text{ are disjoint teams}\}$, and satisfy consistency. More precisely, $\succ' \in \mathcal{C}_{obs}$ if $\succ'$ is a total order on all teams that satisfies consistency and there exist probabilities $P'_{A,B}$ for all pair of teams (A, B) such that $A \succ' B$ iff $P'_{A,B} > 1/2$, and $P'_{A,B} = P_{A,B}$ for all disjoint teams A and B. Lastly, we define $A \succ^* B$ if and only if $A \succ' B$ for all $\succ' \in \mathcal{C}_{obs}$, where A and B are not necessarily disjoint. We refer to $\succ^*$ as the *deducible* relation. For single player relations, we define $a \succ^* b$ if and only if there exists $S \subseteq [n] \setminus \{a, b\}$ such that $S \cup \{a\} \succ' S \cup \{b\}$ for all $\succ' \in \mathcal{C}_{obs}$.

Next, we define two sets of *potential witnesses* that have a simple structure and, in some cases, allow us to deduce single players relation: (1) A *potential subsets witnesses* set, denoted by $\mathcal{S}_{a,b}$, that contains all pairs (S, S') such that S and S' are disjoint subsets of $[n] \setminus \{a, b\}$ and both are of size $k - 1$, and (2) A *potential subset-team witnesses* set, denoted by $\mathcal{T}_{a,b}$, that contains all pairs (S, T) where S and T are disjoint subsets of $[n] \setminus \{a, b\}$, such that S is of size $k-1$ and T is of size k (and is therefore a team). Below, we define under which conditions a potential witnesses is a *witness*.

Definition 3.1. *An element* $(S, S') \in \mathcal{S}_{a,b}$ *is a* subsets witness *for* $a \succ b$ *if* $P_{S\cup\{a\},S'\cup\{b\}} > P_{S\cup\{b\},S'\cup\{a\}}$. *An element* $(S, T) \in \mathcal{T}_{a,b}$ *is a* subset-team witness *for* $a \succ b$ *if* $P_{S\cup\{a\},T} > P_{S\cup\{b\},T}$.

We capture the set of the elements of $\mathcal{S}_{a,b}$ that are subsets witnesses for $a \succ b$ by $\mathcal{S}^*_{a,b}$ and analogously, $\mathcal{T}^*_{a,b} = \{(S,T) \in \mathcal{T}_{a,b} \mid (S,T)$ is a subset-team witness for $a \succ b\}$. It might be the case that $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$ is empty, in particular this holds when $b \succ a$. It is also possible that both $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$ and $\mathcal{S}^*_{b,a} \cup \mathcal{T}^*_{b,a}$ are empty, in which case we will show that the relation between players in a and b cannot be deduced. The following theorem implies that the other direction is also true.

Theorem 3.2. *Let $a, b \in [n]$. Then, $a \succ^* b$ if and only if $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$.*

Proof sketch. Assume that $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$. We show that $a \succ^* b$ by using SST, the fact that $\succ$ is a consistent strict total order, and an exhaustive case analysis. For the sake of illustration we present only one case here, namely, that $(S, S') \in \mathcal{S}^*_{a,b}$ and that both (1) $S \cup \{a\} \succ S' \cup \{b\}$ and (2) $S \cup \{b\} \succ S' \cup \{a\}$ hold. Assume for contradiction that $a \succ^* b$ does not hold. It thus follows that there exists an order, $\succ' \in \mathcal{C}_{obs}$ for which $b \succ' a$ holds. Let $P'_{A,B}$ be the corresponding winning probabilities. Then, using consistency of $\succ'$ and (1) respectively, we get $S \cup \{b\} \succ' S \cup \{a\} \succ' S' \cup \{b\}$ and from SST $P'_{S\cup\{b\},S'\cup\{b\}} \geq P'_{S\cup\{a\},S'\cup\{b\}} > 1/2$. In addition, applying consistency again, it follows that $S \cup \{b\} \succ S' \cup \{b\} \succ S' \cup \{a\}$. Applying SST once more we get $P'_{S\cup\{b\},S'\cup\{a\}} \geq P'_{S\cup\{b\},S'\cup\{b\}} \geq P'_{S\cup\{a\},S'\cup\{b\}}$, a contradiction to $(S, S') \in \mathcal{S}^*_{a,b}$ (since this implies $P_{S\cup\{a\},S'\cup\{b\}} > P_{S\cup\{b\},S'\cup\{a\}}$).

For the other direction we start by defining $\mathcal{D}_a$ as the set of duels (A, B) such that $a \in A$. Moreover, we define a permutation π on the set of teams, which simply exchanges the players a and b when present. We then show that $a \succ b$ implies $P_{A,B} \geq P_{\pi(A),\pi(B)}$ for all $(A, B) \in \mathcal{D}_a$. Moreover, we show that $a \succ^* b$ implies that there exists $(A, B) \in \mathcal{D}_a$ with $P_{A,B} > P_{\pi(A),\pi(B)}$ as follows. Assume not. Then we show that the relation $\succ'$ defined by $A \succ' B$ iff $\pi(A) \succ' \pi(B)$ is included in C_{obs}. However, $a \succ^* b$ implies that for any $S \subseteq [n] \setminus \{a, b\}$ of size $k-1$ it holds that $S \cup \{a\} \succ^* S \cup \{b\}$ which implies (i) $S \cup \{a\} \succ S \cup \{b\}$ as well as

(ii) $S \cup \{a\} \succ' S \cup \{b\}$. Applying the definitions of $\succ'$ and π, statement (ii) implies $S \cup \{b\} = \pi(S \cup \{a\}) \succ \pi(S \cup \{b\}) = S \cup \{a\}$ and hence yields a contradiction to (i). Finally, take some $(A, B) \in \mathcal{D}_a$ with $P_{A,B} > P_{\pi(A),\pi(B)}$. If $b \in B$, then $(A \setminus \{a\}, B \setminus \{b\}) \in \mathcal{S}^*_{a,b}$, otherwise $(A \setminus \{a\}, B) \in \mathcal{T}^*_{a,b}$. ∎

For the sake of brevity, we introduce the set $\mathcal{X}_{a,b}$ which combines the pairs from $\mathcal{S}_{a,b}$ and $\mathcal{T}_{a,b}$ into a set of triples. Formally, $\mathcal{X}_{a,b} = \{(S, S', T) \mid (S, S') \in \mathcal{S}_{a,b}, (S, T) \in \mathcal{T}_{a,b}\}$. We say that (S, S', T) is a witness for $a \succ b$ if $(S, S') \in \mathcal{S}^*_{a,b}$ or $(S, T) \in \mathcal{T}^*_{a,b}$.

3.1.4 Stochastic Setting

In this section we focus on algorithms identifying, with high probability, the top-k team, which is in particular a Condorcet winning team. The main idea is to reduce the dueling teams setting to the classic dueling bandits setting, by which we refer to [134]. To this end we introduce our *gap parameter*, Δ, which intuitively captures how easy it is to prove the relationship between the top-k and the top-$(k+1)$ player. We start by defining, for any element of $\mathcal{X}_{a,b}$, a random variable $X_{a,b}(S, S', T)$ combining the outcomes of the four duels which help determining whether (S, S', T) is a witness for $a \succ^* b$. Formally,

$$X_{a,b}(S, S', T) = \Big(\mathbb{1}[S \cup \{a\} > S' \cup \{b\}] - \mathbb{1}[S \cup \{b\} > S' \cup \{a\}] + \mathbb{1}[S \cup \{a\} > T] - \mathbb{1}[S \cup \{b\} > T]\Big)/2.$$

Observe that, for every $(S, S', T) \in \mathcal{X}_{a,b}$, we have that $\mathbb{E}[X_{a,b}(S, S', T)] > 0$ if and only if (S, S', T) is a witness for $a \succ b$. Moreover, if $\mathbb{E}[X_{a,b}(S, S', T)] = 0$ for every $(S, S', T) \in \mathcal{X}_{a,b}$ then Theorem 3.2 implies that the pairwise relation between players a, b cannot be deduced. Building upon these random variables for fixed elements in $\mathcal{X}_{a,b}$, we define the random variable $X_{a,b}$ by picking a random triplet

$(S, S', T) \in \mathcal{X}_{a,b}$ and returning a realization of $X_{a,b}(S, S', T)$. Combining with the probabilistic method, we obtain the following theorem, which then bring us to the definition of a gap parameter for this problem.

Theorem 3.3. *For every two players $a, b \in [n]$ it holds that $a \succ^* b$ if and only if $\mathbb{E}[X_{a,b}] > 0$.*

Gap parameter: We define our gap parameter by $\Delta := \mathbb{E}[X_{k,k+1}]$. In the following we show that our gap parameter does not just help us to distinguish between the top k and the top $k+1$ player, but also between other players in A_k^* and players from $[n] \setminus A_k^*$. To this end, we show in Lemma 3.4 that strong stochastic transitivity holds for $\mathbb{E}[X_{a,b}]$. For most elements $(S, S', T) \in \mathcal{X}_{a,b}$ it holds that $\mathbb{E}[X_{a,c}(\pi(S), \pi(S'), \pi(T))] \geq \mathbb{E}[X_{a,b}(S, S', T)]$ (and analogously for $X_{b,c}$), where π is a permutation exchanging b and c, but, surprisingly, this is not true in general. By constructing a charging scheme, we can still show that this holds in expectation over all elements of $\mathcal{X}_{a,b}$, and derive a strong stochastic transitivity for distinguishabilities w.r.t. the total order $\succ$ on the players.

Lemma 3.4. *For a triplet of players $a \succ b \succ c$ it holds that*

$$\mathbb{E}[X_{a,c}] \geq \max\{\mathbb{E}[X_{a,b}], \mathbb{E}[X_{b,c}]\}.$$

This also yields the following theorem, which paves the way for our reduction in what follows.

Theorem 3.5. *For any $a, b \in [n]$ such that $a \in A_k^*, b \notin A_k^*$ it holds that $\mathbb{E}[X_{a,b}] \geq \mathbb{E}[X_{k,k+1}] = \Delta$. Thus, if $\Delta > 0$ and a team A it holds that $\mathbb{E}[X_{a,b}] \geq \Delta$ for every $a \in A, b \in [n] \setminus A$, then $A = A^*$.*

The reduction: We now outline the gap-dependent algorithm for the stochastic

setting. The results we have derived in Section 3.1.3 will allow us to deduce, with high probability, whether a distinguishability of a given pair of players is at least Δ, and if so determine which player is the better one. Intuitively, this is done by performing $\mathcal{O}(\frac{1}{\Delta^2})$ team duels.

We will use $\mathbb{E}[X_{a,b}]$ as a proxy for the distinguishability between two single players, a, b, taking advantage of the fact that if their relation is deducible, then $\mathbb{E}[X_{a,b}] \neq 0$ and in this case $\mathbb{E}[X_{a,b}] > 0$ iff $a \succ b$. Similar the to dueling bandits setting, even though $|\mathbb{E}[X_{a,b}]| < \Delta$ for some pairs of players, identifying A_k^* with high probability is possible.

Since we cannot directly sample $X_{a,b}$, we will instead sample uniformly at random a triplet of sets, (S, S', T) from $\mathcal{X}_{a,b}$. Using $(S, S')(\in \mathcal{S}_{a,b})$ and $(S, T)(\in \mathcal{T}_{a,b})$, we can then perform all the duels required for an unbiased sample of $X_{a,b}(S, S', T)$, which is by itself a sampling of $X_{a,b}$. Given any dueling teams instance, we define a dueling bandits instance as follows: for every two players $a, b \in [n]$,

$$P_{a,b} = 1/2 + \mathbb{E}[X_{a,b}]. \tag{3.1}$$

Where $P_{a,b}$ is the probability that a wins in a (singles) duel against b. Clearly, $1 - P_{a,b} = P_{b,a}$. In addition, Theorem 3.3 implies that a is better than b in this dueling bandits instance iff $a \succ^* b$. So whenever a dueling bandits algorithm is asking for a duel query, (a, b), we can make an independent unbiased sample of $X_{a,b}$ by returning a random sampling of $X_{a,b}(S, S', T) + 1/2$ (i.e., $X_{a,b}$ is a Bernoulli random variable with bias $\mathbb{E}[X_{a,b}(S, S', T)] + 1/2$.). In cases where the realization of $X_{a,b}$ is 0 and the algorithm does not consider ties as a valid duel feedback, we can randomly assign a duel winner. We formalize this idea in the sub-procedure *singlesDuel*, that simulates a duel for classical dueling bandits settings using team duels. Notice that, by Lemma

3.4, the probabilities $P_{a,b}$ defined in (3.1) satisfy SST with respect to the total order among the players induced by the ground truth order $\succ$. In addition, the feedback of each single player duel we perform is time-invariant, thus all the non-parametric assumptions for dueling bandits settings apply here. The reduction allows us to identify the top k players using any dueling bandit algorithm with the same goal that works for total order on arms that satisfy SST, and a gap between the top k and $k+1$ arms as assumptions. Formally,

Theorem 3.6. *Given any dueling teams instance with n and k (namely, $P_{A,B}$ for every two teams that hold strict total order, SST, and consistency), we have that the dueling bandit instance defined by (3.1) satisfies SST with respect to the ground truth order among players $\succ$ and for any two players $a \succ b$ it holds that $P_{a,b} \geq 1/2$. Moreover, $P_{k,k+1} = 1/2 + \Delta$.*

Using the above theorem we can use any dueling bandit algorithm for top k identification to solve our problem. [100] provide an algorithm that returns the top k players with probability exceeding $1 - (\log n)^{-c_0}$ with sample complexity at most $c_1(n + k \log k)\frac{\max(\log\log n, \log k)}{\Delta^2_{k,k+1}}$ in expectation, where c_0 and c_1 are universal positive constants and $\Delta_{k,k+1}$ is the distinguishability between the k and the $k+1$ best players (see Algorithm 2 and Theorem 1 in [100]).

[112] show an algorithm that returns the top k players with probability at least $1 - \delta$ with sample complexity $\mathcal{O}(\sum_{i\in[n]}(\Delta_i^{-2}(\log(n/\delta) + \log\log \Delta_i^{-1}))$, where $\Delta_i = \mathbb{1}_{i\succ k+1} \cdot \Delta_{i,k+1} + \mathbb{1}_{k\succ i} \cdot \Delta_{k,i}$ and $k, k+1$ are the top k and the top $k+1$ players, respectively (see Algorithm 5 and Theorem 8 in [112])[3].

These algorithms, together with Theorem 3.3 allow us to derive the following theorem.

[3]We remark that [112] also assume Stochastic triangle inequality which we do not, however it is only used to derive a lower bound.

Theorem 3.7. *There exists an algorithm that returns A_k^* with probability exceeding $1-(\log n)^{-c_0}$ with sample complexity at most $c_1(n+k\log k)\frac{\max(\log\log n,\log k)}{\Delta^2}$ in expectation, where c_0 and c_1 are universal positive constants.*

In addition, there exists an algorithm that returns A_k^* with probability at least $1-\delta$ with sample complexity $\mathcal{O}(\sum_{i\in[n]}(\Delta_i^{-2}(\log(n/\delta)+\log\log\Delta_i^{-1}))$, where $\Delta_i = \mathbb{1}_{i\succ k+1}\cdot \mathbb{E}[X]_{i,k+1} + \mathbb{1}_{k\succ i}\cdot\mathbb{E}[X]_{k,i}$ and i denotes the top i players, thus $\Delta_i \geq \Delta$ for every $i\in[n]$.

3.1.5 Deterministic Setting

In the previous section we showed the existence of algorithms that identify the top k team with a number of duels that depends on Δ. But what if Δ is very small or even 0? One reason for that can be that all relevant probabilities are close to 1/2. More precisely, $P_{\{k\}\cup S,\{k+1\}\cup S'}$, $P_{\{k+1\}\cup S,\{k\}\cup S'}$, $P_{\{k+1\}\cup S,T}$, and $P_{\{k\}\cup S,T}$ are very close to 1/2 for all $(S,S',T)\in\mathcal{X}_{k,k+1}^*$. This might also occur in classic dueling bandits settings, when the target is to separate the top k players from the rest (e.g., [100, 112]). As a result, a gap between the top k and $k+1$ players is often a parameter of the sample complexity in such settings. For these cases, our approach presented in the stochastic section very much resembles the current literature.

The other, more interesting reason for Δ to be small is when there exist only a small number of witnesses. This is in particular the case when the probability matrix contains only few distinct values, as for example when feedback is deterministic, i.e., $P_{A,B}\in\{0,1\}$. Note that in this setting, $(S,T)\in\mathcal{T}_{a,b}$ is a witness if and only if $S\cup\{a\}\succ_{obs} T\succ_{obs} S\cup\{b\}$, and $(S,S')\in\mathcal{S}_{a,b}$ is a witness if and only if $S\cup\{a\}\succ_{obs} S'\cup\{b\}$ and $S'\cup\{a\}\succ_{obs} S\cup\{b\}$. This follows as for any other potential witness $(S,S',T)\in\mathcal{X}_{a,b}$ it holds that $\mathbb{E}[X_{a,b}(S,S',T)]=0$. It is possible to come up with deterministic instances where up to $(2k-1)^2$ pairs do not have any witness to

distinguish them. To overcome this issue, we design algorithms for the deterministic case that are independent of Δ within this section. In the Section 3.1.8 we show that these results can be extended to a slightly stochastic environment.

The limitation of the set of witnesses makes the problem of identifying a Condorcet winning team in the deterministic setting surprisingly nontrivial. For general total orders, a crucial difficulty lies in efficiently proving that a given team is indeed Condorcet winning. However, we are still able to get the following result:

Theorem 3.8. *For deterministic feedback, there exists an algorithm that performs* $\mathcal{O}(kn\log(k) + k^2\log(k)2^{5k})$ *duels and outputs a Condorcet winning team.*

For the natural special case of *additive total orders* we obtain a stronger result. A total order $\succ$ is *additive total*, if there exist values for the players denoted by $v(a), a \in [n]$ such that $A \succ B$ iff $\sum_{a\in A} v(a) > \sum_{b\in B} v(b)$. We present an algorithm that identifies a Condorcet winning team after polynomial many duels and also outputs a proof.

Theorem 3.9. *For deterministic feedback and additive total orders, there exists an algorithm that finds a Condorcet winning team within* $\mathcal{O}(kn\log(k) + k^5)$ *duels.*

Both algorithms rely on the same preprocessing procedure called *ReducePlayers* which reduces the number of players from n to $\mathcal{O}(k)$. At the heart of this procedure is a subroutine called *Uncover*. After describing *Uncover* and *ReducePlayers*, we prove Theorem 3.8. Towards proving Theorem 3.9, we introduce two more subroutines, namely *NewCut* and *Compare*, which are crucial for identifying and proving a Condorcet winning team within the smaller instance. Finally, Algorithm *CondorcetWinning* combines all components and proves Theorem 3.9.

The Uncover Subroutine Given two disjoint teams $A \succ B$, the *Uncover* subroutine finds a pair of players $a \in A$ and $b \in B$ and a subsets witness for their relation,

i.e., an element from $\mathcal{S}^*_{a,b}$. To understand the idea of the subroutine, consider some arbitrary ordering of the elements in A and B, respectively, i.e., $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$. Then, iteratively exchange the elements a_1 and b_1, a_2 and b_2, resulting in sets $A_0 = A, B_0 = B, A_1 = \{b_1, a_2, \ldots, a_k\}, B_1 = \{a_1, b_2, \ldots, b_k\}$, $A_2 = \{b_1, b_2, a_3, \ldots, a_k\}$, and so on. Since $A_0 \succ B_0$ but $A_0 = B_k \succ A_k = B_0$ holds, there needs to be some earliest point in time $i \leq k$ for which $B_i \succ A_i$ is true. This implies $a_i \succ b_i$ as $(\{a_1, \ldots a_{i-1}, b_{i+1}, \ldots b_k\}, \{b_1, \ldots, b_{i-1}, a_{i+1}, \ldots, a_k\})$ is a witness for this relation.

While the above sketched subroutine is simple, it performs k duels in the worst case. We refine this idea by a binary search approach, decreasing the number of duels to $\log(k)$. In Section 3.1.8 we give a slightly stronger version of Lemma 3.10, which allows us to divide A and B into two subsets each. Under some requirements, we can then control from which of the subsets the pairwise relation is revealed.

Lemma 3.10. *Let A and B be two disjoint teams with $A \succ B$. After performing $\mathcal{O}(\log(k))$ duels,* Uncover *returns (a, b) with $a \in A$, $b \in B$ and $(S, S') \in \mathcal{S}^*_{a,b}$.*

Reducing the Number of Players to $\mathcal{O}(k)$ The fact that we can eliminate a subset of the players and still find (and prove) a Condorcet winning team is due to the following observation.

Observation 3.11. *Let $X \subseteq [n]$ such that $A^*_{2k} \subseteq X$. Let $\hat{A} \subseteq X$ be a team such that $\hat{A} \succ A$ for all teams $A \subseteq X \setminus \hat{A}$. Then, $\hat{A}$ is a Condorcet winning team.*

The procedure *ReducePlayers* reduces the set of players $[n]$ to some subset $X \subseteq [n]$ guaranteeing that $A^*_{2k} \subseteq X$ and $|X| < 6k$. The algorithm maintains a dominance graph $D = (V, E)$ on the set of players. More precisely, the nodes of D are the players, i.e., $V = [n]$, and there exists an arc from node a to node b if the algorithm has proven

that $a \succ b$. The set $V_{<2k}$ is the subset of the players having an indegree smaller than $2k$ in D. The high level idea of the algorithm is the following: It starts with the empty dominance graph $D = ([n], \emptyset)$. The algorithm then iteratively identifies pairwise relations of the players with help of *Uncover* and adds the corresponding arcs to the graph. By adding more and more arcs to D, the set of nodes $V_{<2k}$ shrinks more and more while $A^*_{<2k} \subseteq V_{<2k}$ is always guaranteed. At some point, the algorithm cannot identify any more pairwise relations and returns $V_{<2k}$. How does the algorithm identify pairwise relations? At any point it tries to find a matching between $2k$ players, say $\{(a_1, b_1), \ldots, (a_k, b_k)\}$ with the constraint that, for all $i \in [k]$, none of the arcs (a_i, b_i) or (b_i, a_i) is present within the graph D yet. The algorithm ends when it cannot find such a matching anymore. We show that this only happens after $|V_{<2k}| < 6k$.

Lemma 3.12. *Given the set of players* $[n]$*,* ReducePlayers *returns* $X \subseteq [n]$ *with* $|X| \leq 6k - 2$ *and* $A^*_{2k} \subseteq X$*.* ReducePlayers *performs* $\mathcal{O}(nk \log(k))$ *duels and runs in time* $\mathcal{O}(n^2k^2)$*.*

Proof Sketch (of Theorem 3.8). Let D be the dominance graph at the end of *ReducePlayers*. Then, the learner selects a k-sized subset of $V_{<2k}$, call it $\hat{A}$, with the property that there is no arc from some node in $V_{<2k} \setminus \hat{A}$ towards some node in $\hat{A}$. Then, the learner tests $\hat{A}$ against all possible teams containing players from $V_{<2k} \setminus \hat{A}$, which are $\mathcal{O}(2^{5k})$ many. If $\hat{A}$ wins all of these duels, then $\hat{A}$ is a Condorcet winning team by Observation 3.11. However, if there exists $A \succ \hat{A}$, then, by the choice of $\hat{A}$, there does not exist any arc from A towards $\hat{A}$. Hence, by calling the subroutine *Uncover* for two arbitrary orderings of A and $\hat{A}$, the learner will identify one additional arc. This procedure can be repeated $\mathcal{O}(k^2)$ many times and hence shows Theorem 3.8. ∎

Subroutines NewCut and Compare The *NewCut* subroutine takes as input a subset of the players $X \subseteq [n]$, a pair $a, b \in X$, and a witness proving that $a \succ b$, i.e., $(S, T) \in \mathcal{S}^*_{ab} \cup \mathcal{T}^*_{ab}$. That means, T can be either of size $k-1$ or k, and S and T are not required to be subsets of X. The subroutine outputs a partition of X into two non-empty sets U and L with $U \triangleright L$, which is short-hand notation for $u \succ \ell$ for any $u \in U$ and $\ell \in L$. The main idea of the algorithm is to make use of the transitivity among witnesses which we established in Lemma 3.4. More precisely, whenever the algorithm has found a witness showing that some player in X is better than b, it stores this witness in a list $\mathcal{W}$. Iteratively applying the transfer function used in the proof of Lemma 3.4 to all witnesses in $\mathcal{W}$ and elements in X makes sure that after termination of the algorithm $U \triangleright L$ holds, where U is defined by all players u for which the algorithm found a witness for $u \succ b$ and $L = X \setminus U$.

Lemma 3.13. *Let $X \subseteq [n]$, $a, b \in X$ and $(S, T) \in \mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$. Then,* NewCut$(X, (a, b), (S, T))$ *returns a partition of X into U and L such that $U \triangleright L$, $a \in U$ and $b \in L$. The number of duels performed by* NewCut *and its running time can be bounded by $\mathcal{O}(|X|^2)$.*

From now on we assume additive linear orders. The *compare* subroutine is crucial for obtaining upper bounds for differences of values of subsets of players. It is used in the following situation. Let (a, b) be a pair of players and $(S, S') \in \mathcal{S}^*_{a,b}$ be a witness for $a \succ b$. Then, it can be easily shown that $v(a) - v(b) > |v(S) - v(S')|$. We will be interested in the question whether a similar relation holds for two subsets of S and S', namely, $C \subseteq S$ and $D \subseteq S'$ of equal size. The *compare* subroutine checks whether such a relation holds by performing two additional duels. If it returns *True*, then $v(a) - v(b) > |v(C) - v(D)|$. Otherwise, there can be found a pair $c \in C$ and $d \in D$ and a witness for their relation by one call to the *Uncover* subroutine. This observation is formalized Lemma 3.14.

Lemma 3.14. *Let $a \succ b$ be two players, $(S, S') \in \mathcal{S}^*_{a,b}$ and $C \subseteq S, D \subseteq S'$ with $|C| =$*

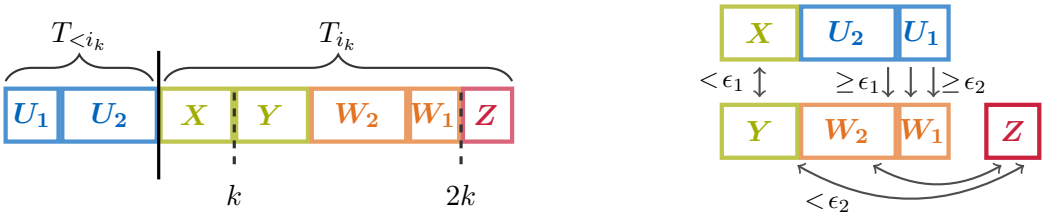


Figure 3.1: Illustration of the proof technique of algorithm *CondorcetWinning1*. In the left illustration, the solid black line indicates that all players left to it were proven to be better than all players right to it. The dashed line marked with "k" indicates that the sets to its left contain k players in total. However, this line does not indicate proven relations, e.g., a player from X can be better than a player from Y. The right figure illustrates the proof for $X \cup U_1 \cup U_2$ being Condorcet winning.

$|D|$. *If* Compare$((a,b),(S,S'),(C,D))$ *returns* True*, then* $v(a)-v(b) > |v(C)-v(D)|$. *Otherwise, one call to* Uncover *returns* $c \in C$ *and* $d \in D$ *together with a witness for their relation.*

Algorithm CondorcetWinning The algorithm maintains a partition of the players into a weak ordering, i.e., $\mathcal{T} = \{T_1, \dots, T_\ell\}$ with $T_1 \triangleright T_2 \triangleright \cdots \triangleright T_\ell$. We introduce the short-hand notation $T_{\leq j} = \bigcup_{m\in[j]} T_m$ and $T_{<j} = \bigcup_{m\in[j-1]} T_m$. After the application of the preprocessing procedure *ReducePlayers*, this partition consists of one set, namely $\mathcal{T} = \{T_1\}$, where $|T_1| \in \mathcal{O}(k)$ and $A^*_{2k} \subseteq T_1$. At any point in the execution of the algorithm, we are especially interested in two indices, namely $i_k \in [\ell]$ such that $|T_{<i_k}| < k < |T_{\leq i_k}|$ and similarly $i_{2k} \in [\ell]$ such that $|T_{<i_{2k}}| < 2k < |T_{\leq i_{2k}}|$. [4] Observe that all players from $T_{<i_k}$ are guaranteed to be among the top-k players. On the other hand, among the players from T_{i_k} some belong to A^*_k and others do not. The main idea of the algorithm is then the following: Take a prefix of $\mathcal{T}$ of size k, i.e., this team contains the set of players $T_{<i_k}$ and is a subset of the players in $T_{\leq i_k}$, and

[4] In case one of these indices does not exist, this implies that we have either identified the set A^*_k or A^*_{2k}. In the first case we have found a Condorcet winning team and in the second case Observation 3.11 implies that we can find one by performing one additional duel. For the sake of brevity we disregard this case from now on.

either prove that this prefix is a Condorcet Winning team, or refine the partition $\mathcal{T}$ and repeat the process.

We provide two different algorithms, namely *CondorcetWinning1* for the case $i_k = i_{2k}$ and *CondorcetWinning2* when $i_k \neq i_{2k}$. Unsurprisingly, the latter case requires a strictly less sophisticated approach, which is why we focus on *CondorcetWinning1* in the following.

The algorithm starts by partitioning the set $T_{<i_k}$ into two sets U_1 and U_2, where U_1 is a prefix of $T_{<i_k}$ of size $|T_{\leq i_k}| - 2k$. It partitions the set T_{i_k} into five sets X, Y, W_1, W_2, and Z. In particular it is known that $(U_1 \cup U_2) \triangleright (X \cup Y \cup W_1 \cup W_2 \cup Z)$ but no relation among any pair in T_{i_k} is known. Regarding the sizes of the sets it holds that $|U_i| = |W_i|$ for $i \in \{1, 2\}$, $|X| = |Y| = k - |U_1| - |U_2|$ and $|U_1| = |Z|$. The main aim of the algorithm will be to define $0 < \epsilon_1 < \epsilon_2$ and prove that the following statements are true:

(i) $|v(X) - v(Y)| < \epsilon_1$

(ii) $|v(a) - v(b)| < \epsilon_2$ for all $a \in Y \cup W_1 \cup W_2$ and $b \in Z$, and

(iii) there exist $u_1, \ldots, u_{|Z|+1} \in U_1 \cup U_2$ as well as $w_1, \ldots, w_{|Z|+1} \in W_1 \cup W_2$ such that

  (a) $v(u_1) - v(w_1) \geq \epsilon_1$ and

  (b) $v(u_i) - v(w_i) \geq \epsilon_2$ for all $i \in \{2, \ldots, |Z| + 1\}$.

With these three statements we can show that $U_1 \cup U_2 \cup X$ is a Condorcet winning team. More precisely, one can show that $v(U_1 \cup U_2 \cup X) - v(W_1 \cup W_2 \cup Y) > |Z| \cdot \epsilon_2$ and $v(W_1 \cup W_2 \cup Y) - v(B^*) > -|Z| \cdot \epsilon_2$, where B^* is the best response towards $U_1 \cup U_2 \cup X$. See Figure 3.1 for an illustration of the argument.

It remains to sketch how the algorithm defines ϵ_1, ϵ_2 and proves $(i) - (iii)$. For

simplicity assume $U_1 \triangleright U_2$. The algorithm then attempts to do the following steps: (1) Find a witness for players $\bar{u} \in U_2$ and $\bar{w} \in W_2$, using *Uncover*. (2) Use *Compare*, to prove that $|v(X) - v(Y)| < v(\bar{u}) - v(\bar{w})$ and $|v(a) - v(b)| < v(\bar{u}) - v(\bar{w})$ holds for all players $a \in W_1 \cup W_2 \cup Y$ and $b \in Z$. (3) Repeat step (2) by replacing $\bar{w}$ with any player of W_1. If one of the steps (1)-(3) fails, we show that the partition $\mathcal{T}$ can be refined. Otherwise, we show that $(i) - (iii)$ hold for $\epsilon_1 = v(\bar{u}) - v(w_1^*)$ and $\epsilon_2 = v(\bar{u}) - v(w_2^*)$, where w_1^* and w_2^* are the best and second best players from $W_1 \cup \{\bar{w}\}$, respectively. The following Lemma concludes the proof sketch of Theorem 3.9.

Lemma 3.15. *After performing $\mathcal{O}(k^5)$ many duels,* CondorcetWinning1 *has identified a Condorcet winning team.* CondorcetWinning2 *identifies a Condorcet winning team after $\mathcal{O}(k^2 \log(k))$ duels.*

3.1.6 Extended Version and Proofs of Section 3.1.3

Within the main text, we covered two different types of witnesses for single players relations. In this section, we show that whenever a relation between single players can be proven from observable duels in our setting, there exists at least one type of witness for it. For the convince of the reader, we recall the definitions mentioned in the main text in a comprehensive manner, provide more explanations and some examples.

Possible Witnesses For two players a and b we define $\mathcal{S}_{a,b}$ as the set of pairs of disjoint $k-1$ sized subsets of players from $[n] \setminus \{a, b\}$, i.e.,

$$\mathcal{S}_{a,b} = \{(S, S') \mid S, S' \subseteq [n] \setminus \{a, b\}, S \cap S' = \emptyset, |S| = |S'| = k-1\},$$

and $\mathcal{T}_{a,b}$ as the set of disjoint $k-1$ sized subset S and a team T pair from $[n] \setminus \{a, b\}$, i.e.,

$$\mathcal{T}_{a,b} = \{(S, T) \mid S, T \subseteq [n] \setminus \{a, b\}, S \cap T = \emptyset, |S| = k-1, |T| = k\}.$$

Definition 3.16 (Witnesses and Witnesses sets)**.** *A* witness for $a \succ b$ *is one of the following types:*

(i) Subsets*: A pair of disjoint subsets* $(S, S') \in \mathcal{S}_{a,b}$ *such that*

$$P_{\{a\}\cup S, \{b\}\cup S'} > P_{\{b\}\cup S \succ \{a\}\cup S'}.$$

We denote the set of all subsets witnesses for $a \succ b$ *by* $\mathcal{S}^*_{a,b}$.
(ii) Subset-Team*:* $(S, T) \in \mathcal{T}_{a,b}$*, such that*

$$P_{\{a\}\cup S, T} > P_{\{b\}\cup S, T}.$$

We denote the set of all subset-team witnesses for $a \succ b$ *by* $\mathcal{T}^*_{a,b}$.

In case we find a witness, we can use it to compare players as follows.

Lemma 3.17. *If there exists a pair* $(S, S') \in \mathcal{S}^*_{a,b}$*, or a pair* $(S, T) \in \mathcal{T}^*_{a,b}$*, then* $a \succ b$*.*

Proof. First, consider the existence of $(S, S') \in \mathcal{S}^*_{a,b}$.
Hence

$$(*)\ P_{S\cup\{a\},S'\cup\{b\}} > P_{S\cup\{b\},S'\cup\{a\}}$$

Assume for contradiction that $b \succ a$. Consistency implies $S \cup \{b\} \succ S \cup \{a\}$ and $S' \cup \{b\} \succ S' \cup \{a\}$.

Adding up the two implications from the witness definition and SST, we have

$$P_{S\cup\{a\},S'\cup\{b\}} >_{(*)} P_{S\cup\{b\},S'\cup\{a\}} >_{b\succ a} P_{S\cup\{a\},S'\cup\{a\}} >_{b\succ a} P_{S\cup\{a\},S'\cup\{b\}},$$

Which is a contradiction.

Now, consider the existence of $(S, T) \in \mathcal{T}^*_{a,b}$. We have that

$$(**)\ P_{S\cup\{a\},T} > P_{S\cup\{b\},T}$$

Assume for contradiction that $b \succ a$. Consistency implies $S \cup \{b\} \succ S \cup \{a\}$.

$$P_{S\cup\{b\},T} >_{b\succ a} P_{S\cup\{a\},T} >_{(**)} P_{S\cup\{b\},T},$$

Which is a contradiction. ∎

Note that while the above lemma implies a sufficient condition for $a \succ b$, there is no guarantee that for every $a \succ b$ there exists a witness that proves it, as it requires disjoint subsets. For example, consider a lexicographical order among teams with

$n = 4, k = 2$ with uniform noise, e.g. when $P_{A,B} = 0.6$ for all teams $A \succ B$. It follows from consistency and $12 \succ 23$ that $2 \succ 3$, but there is no witness for that. Moreover, even if we execute each of the 3 possible duels enough to estimate correctly that $P_{12,34} = P_{13,24} = P_{14,23} = 0.6$ there is no way to distinguish between the second and third best players. In what follows we formalize this intuition, showing that if single players relation is provable then one of the aforementioned witnesses types exists for it.

Next, we recall the Observable relation and the set $\mathcal{C}_{obs}$.

Observable relation Let $\succ_{obs}$ denote the relation between every two disjoint teams, i.e.,

$$A \succ_{obs} B \iff A \succ B,\ |A| = |B| = k,\ A \cap B = \emptyset,\ A, B \subseteq [n].$$

Namely the relation $\succ_{obs}$ is deducible from valid duels [5].

In what follows, we elaborate more on the definition of $\mathcal{C}_{obs}$ by defining first a set for Compatible winning probabilities.

Compatible winning probabilities Let $\mathbb{P}_{obs}$ be the set of all tuples $(P', \succ')$, where P' are the winning probability matrices for teams, i.e., $P' = (P'_{A,B})_{A \neq B, |A|=|B|=k, A,B \in [n]} \in [0,1]^{\binom{n}{k}} \times [0,1]^{\binom{n}{k}}$, and $\succ'$ is a total order on the teams such that:

1. For every pair of disjoint teams (A, B) the winning probability matrix P' has the same winning probability as the ground truth P, i.e., $A \cap B = \emptyset$ implies $P'_{A,B} = P_{A,B}$.
2. It holds that $P'_{A,B} = 1/2$ iff $A = B$.
3. $P'_{A,B} > 1/2$ if and only if $A \succ' B$.

[5] Notice that technically, $\succ_{obs}$ is not defined on pairs of different teams which are not disjoint, and therefore not even a partial order on teams (e.g., we have that $\{a, b\} \succ_{obs} \{c, d\} \succ_{obs} \{a, e\}$ but $\{a, b\} \not\succ_{obs} \{a, e\}$ as they share a player and the duel $(\{a, b\}, \{a, e\})$ is not observable.).

4. P' satisfies SST w.r.t. $\succ'$.

Namely, $\mathbb{P}_{obs}$ contains all tuples $(P', \succ')$ that do not contradict the winning probabilities the learner can observe and our assumptions.

Compatible relations Let $\mathcal{C}_{obs}$ be the set of all total orders $\succ'$ for which there exists $(P', \succ') \in \mathbb{P}_{obs}$. Notice that by the definition of $\mathbb{P}_{obs}$, we know that $\succ'$ satisfy consistency and in particular it holds that $A \succ' B$ for every disjoint teams (A, B) with $A \succ_{obs} B$. Namely, $\mathcal{C}_{obs}$ is the sets of all possible total orders that could explain the results of the observable duels.

We remark that it follows directly from the definition of $\mathbb{P}_{obs}$ that $(P, \succ) \in \mathbb{P}_{obs}$, where P is the ground truth winning probability matrix and $\succ$ the ground truth total order. Because of this, it also holds that $\succ$ is in $\mathcal{C}_{obs}$. To illustrate that $\succ$ is typically not the only total order in $\mathcal{C}_{obs}$, we provide the following example.

Example 3.18. *For $n = 5, k = 2$, consider the lexicographic order , i.e., $\{1,2\} \succ \{1,3\} \succ \{1,4\} \succ \{1,5\} \succ \{2,3\} \succ \{2,4\} \succ \{2,5\} \succ \{3,4\} \succ \{3,5\} \succ \{4,5\}$ and assume $P_{A,B} = 0.6$ iff $A \succ B$ (equivalently $P_{A,B} = 0.4$ iff $B \succ A$). Then, we have that*

$$A \succ_{obs} B \iff \begin{cases} 1 \in A, \text{ or} \\ 1 \notin A \cup B, 2 \in A. \end{cases}$$

While $\succ \in \mathcal{C}_{obs}$, there are other consistent total orders in $\mathcal{C}_{obs}$, such as $\{1,2\} \succ' \{1,5\} \succ' \{1,4\} \succ' \{1,3\} \succ' \{2,5\} \succ' \{2,4\} \succ' \{2,3\} \succ' \{5,4\} \succ' \{5,3\} \succ' \{4,3\}$ (the order $\succ'$ is obtained by swapping players 3 and 5 in $\succ$). Similarly, the probability matrices $P_{A,B} = 0.6$ for all $A \succ B$, (the ground truth), but $P^1_{A,B} = 0.7 \; \forall \; A \succ B$ and $P^2_{A,B} = 0.6 \; \forall \; A \succ' B$ are also in $\mathbb{P}$.

We now recall the definition of the deducible relation, $\succ^*$ for both teams and single

players, where the latter definition is a combination of the former and single players consistency.

The intuition behind these definitions is that a relation can be deducible (proven) by team duels if any "reasonable" total order that could possibly be the ground order agree on this relation. We stress that both $\mathbb{P}_{obs}$ and $\mathcal{C}_{obs}$ are strictly for analysis, as we do not need to explicitly calculate them.

Definition 3.19. *Team A is deducibly better than a different team B, denoted by $A \succ^* B$ (using team duels), if $A \succ' B$ for all $\succ' \in \mathcal{C}_{obs}$.*

Definition 3.20. *Player a is deducibly better than player b, denoted by $a \succ^* b$, if $\{a\} \cup S \succ' \{b\} \cup S$ for all $\succ' \in \mathcal{C}_{obs}$.*

We continue with an example for relations that $\succ^*$ must satisfy. Suppose the learner has observed that $\{a, c\} \succ_{obs} \{b, d\} \succ_{obs} \{a, e\} \succ_{obs} \{c, d\}$. Since all the relations $\succ \in \mathcal{C}_{obs}$ satisfy transitivity, it follows that $\{b, d\} \succ^* \{c, d\}$, $\{a, c\} \succ^* \{a, e\}$, and $\{a, c\} \succ^* \{c, d\}$. As each $\succ \mathcal{C}_{obs}$ also satisfies single players consistency, we deduce $b \succ^* c$, $c \succ^* e$ and $a \succ^* d$, respectively. Applying single players consistency again, we can get, for example, $\{a, b\} \succ^* \{a, c\} \succ^* \{a, e\} \succ^* \{d, e\}$ (using $b \succ^* c$, $c \succ^* e$ and $a \succ^* d$, respectively).

Intuitively, what we will show in Theorem 3.2 is that for every pair of players that one is provably better than the another there exists a witness for it, thus there is a short proof with which the learner can verify their relation with $O(1)$ queries in the deterministic case. Before we start proving the Theorem 3.2 we prove the following helpful lemma.

Lemma 3.21. *Let $\succ \in C_{obs}$ and P be a corresponding probability matrix satisfying SST.*

Let $a, b \in [n]$ with $a \succ b$. Then, the following holds true:

1. *Let $(S, S') \in \mathcal{S}_{a,b}$, then $P_{\{a\}\cup S,\{b\}\cup S'} \geq P_{\{b\}\cup S,\{a\}\cup S'}$.*
2. *Let $(S, T) \in \mathcal{T}_{a,b}$, then $P_{\{a\}\cup S,T} \geq P_{\{b\}\cup S,T}$.*

Proof. 1. We start by proving that for every $(S, S') \in \mathcal{S}_{a,b}$ it holds that

$$P_{\{a\}\cup S,\{b\}\cup S'} \geq P_{\{b\}\cup S,\{a\}\cup S'}$$

by exhaustion.

(a) If $S \cup \{a\} \succ S' \cup \{b\}$ and $S' \cup \{a\} \succ S \cup \{b\}$ then it follows that $1/2 < P_{\{a\}\cup S,\{b\}\cup S'}, P_{\{a\}\cup S',\{b\}\cup S}$ and therefore

$$P_{\{a\}\cup S,\{b\}\cup S'} > 1/2 > 1 - P_{\{a\}\cup S',\{b\}\cup S} = P_{\{b\}\cup S,\{a\}\cup S'}.$$

(b) If (a) does not hold, then it follows that either of the following holds true:

(i) $\{b\} \cup S \succ \{a\} \cup S'$ (and $\{a\} \cup S \succ \{b\} \cup S'$ as $b \not\succ a$).
From single players consistency of $\succ$ we have that

$$\{a\} \cup S \succ \{b\} \cup S \succ \{a\} \cup S' \succ \{b\} \cup S'$$

Applying SST, we have that

$$P_{\{a\}\cup S,\{b\}\cup S'} \geq P_{\{a\}\cup S,\{a\}\cup S'} \geq P_{\{b\}\cup S,\{a\}\cup S'}.$$

(ii) $\{b\} \cup S' \succ \{a\} \cup S$ (and $\{a\} \cup S' \succ \{b\} \cup S$ as $b \not\succ a$).

From consistency, we have that

$$\{a\} \cup S' \succ \{b\} \cup S' \succ \{a\} \cup S \succ \{b\} \cup S$$

Applying SST, we have that

$$P_{\{a\}\cup S',\{b\}\cup S} \geq P_{\{a\}\cup S',\{a\}\cup S} \geq P_{\{b\}\cup S',\{a\}\cup S}.$$

Therefore

$$1 - P_{\{b\}\cup S',\{a\}\cup S} \geq 1 - P_{\{a\}\cup S',\{b\}\cup S}.$$

Applying $P_{A,B} = 1 - P_{B,A}$ for every $A, B \in [n]$,

$$P_{\{a\}\cup S,\{b\}\cup S'} \geq P_{\{b\}\cup S,\{a\}\cup S'}.$$

(iii) The case that $\{b\} \cup S' \succ \{a\} \cup S$ and $\{b\} \cup S' \succ \{a\} \cup S$ cannot hold as it would imply $b \succ a$ which is a contradiction to $a \succ b$, as $\succ$ being a consistent total order yields a total order on players.

2. Strict total order on teams together with consistency implies that either of the following holds: (a) $\{a\} \cup S \succ \{b\} \cup S \succ T$, (b)$\{a\} \cup S \succ T \succ \{b\} \cup S$, or (c) $T \succ \{a\} \cup S \succ \{b\} \cup S$. Applying SST on (a) and (c) proves the claim, and if (b) holds we have

$$P_{\{a\}\cup S,T} > 1/2 > P_{\{b\}\cup S,T}.$$

■

We note that the left to right direction in the following sentence is very similar to Lemma 3.17 and their proofs are equivalent, however for completeness we provide a full proof here as well.

Theorem 3.2. *Let $a, b \in [n]$. Then, $a \succ^* b$ if and only if $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$.*

Proof. We start with the direction from right to left, i.e., $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$ implies $a \succ^* b$.

First, consider $(S, S') \in \mathcal{S}^*_{a,b}$ and assume for contradiction that $a \succ^* b$ does not hold. That is, there exists $\succ' \in \mathcal{C}_{obs}$ and $P' \in \mathbb{P}_{obs}$ such that $b \succ' a$, and P' is a corresponding winning probability matrix.

By Lemma 3.21 and the definition of $\mathbb{P}_{obs}$ it follows that

$$P_{S\cup\{b\},S'\cup\{a\}} = P'_{S\cup\{b\},S'\cup\{a\}} \geq P'_{S\cup\{a\},S'\cup\{b\}} = P_{S\cup\{a\},S'\cup\{b\}}$$

holds, as the teams are disjoint. This is a contradiction to $(S, S') \in \mathcal{S}^*_{a,b}$.

Similarly, let $(S, T) \in \mathcal{T}^*_{a,b}$ and assume for contradiction that $a \succ^* b$ does not hold. That is, there exists $\succ' \in \mathcal{C}_{obs}$ and $P' \in \mathbb{P}_{obs}$ such that $b \succ' a$, and P' is a corresponding winning probability matrix. By Lemma 3.21 and the definition of $\mathbb{P}_{obs}$ it follows that

$$P_{S\cup\{b\},T} = P'_{S\cup\{b\},T} \geq P'_{S\cup\{a\},T} = P_{S\cup\{a\},T}$$

holds, as the teams are disjoint. This is a contradiction to $(S, T) \in \mathcal{T}^*_{a,b}$.

We turn to the direction from left to right, i.e. that $a \succ^* b$ yields $S^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$. We start by defining $\mathcal{D}_a$ as the set of observable duels (A, B) such that $a \in A$. Moreover, we define a permutation π on the set of players, which simply exchanges the players

a and b when present. More precisely,

$$\pi(S) = \begin{cases} S \setminus \{a\} \cup \{b\} & \text{if } a \in S, b \notin S \\ S \setminus \{b\} \cup \{a\} & \text{if } b \in S, a \notin S \\ S & \text{else.} \end{cases}$$

We claim that $a \succ^* b$ implies

$$P_{A,B} \geq P_{\pi(A),\pi(B)} \text{ for all } (A,B) \in \mathcal{D}_a \tag{3.2}$$

(P is the ground truth winning probability matrix). To see why, we first define

$$\begin{aligned} \mathcal{D}_a^1 &= \{(A,B) \in \mathcal{D}_a \mid b \in A\} \\ \mathcal{D}_a^2 &= \{(A,B) \in \mathcal{D}_a \mid b \in B\} \\ \mathcal{D}_a^3 &= \{(A,B) \in \mathcal{D}_a \mid b \notin A \cup B\}. \end{aligned}$$

Notice that

$$\mathcal{D}_a = \mathcal{D}_a^1 \cup \mathcal{D}_a^2 \cup \mathcal{D}_a^3 \tag{3.3}$$

When $(A,B) \in \mathcal{D}_a^1$, then $(\pi(A), \pi(B)) = (A,B)$ and $P_{A,B} = P_{\pi(A),\pi(B)}$.

When $(A,B) \in \mathcal{D}_a^2$, then $(A \setminus \{a\}, B \setminus \{b\}) \in \mathcal{S}_{a,b}$, and $P_{A,B} \geq P_{A\setminus\{a\}\cup\{b\},B\setminus\{b\}\cup\{a\}} = P_{\pi(A),\pi(B)}$ follows from Lemma 3.21.

Similarly, when $(A,B) \in \mathcal{D}_a^3$ then $(A \setminus \{a\}, B) \in \mathcal{T}_{a,b}$ and $P_{A,B} \geq P_{A\setminus\{a\}\cup\{b\},B} = P_{\pi(A),\pi(B)}$ follows from Lemma 3.21.

We will now show that $a \succ^* b$ implies the existence of $(A, B) \in \mathcal{D}_a$ with $P_{A,B} > P_{\pi(A),\pi(B)}$.

Assume not. Then in particular from (3.2) we have that $P_{A,B} = P_{\pi(A),\pi(B)}$ holds for all $(A, B) \in \mathcal{D}_a$.

Claim. *Let $\succ'$ be the relation defined by $A \succ' B$ iff $\pi(A) \succ \pi(B)$ with the corresponding winning probabilities defined by $P'_{A,B} = P_{\pi(A),\pi(B)}$. If $P_{A,B} = P_{\pi(A),\pi(B)}$ for every $(A, B) \in \mathcal{D}_a$ then $P' \in \mathbb{P}_{obs}$ and thus $\succ' \in C_{obs}$.*

Proof. Observe that $P_{A,B} = P'_{A,B}$ for all disjoint teams A and B follows by definition. In addition, since π is invertible and involuntary, for every team A there exists a team A_π such that $\pi(A_\pi) = A$ hence $P_{A,A} = P_{A_\pi,A_\pi} = 1/2$. It remains to show that (1) Every pair of different teams A, B holds $P'_{A,B} > 1/2$ iff $A \succ' B$, (2) that $\succ'$ is a total ordering satisfying single players consistency, and (3) that P' satisfy SST w.r.t. $\succ'$.

(1) Let A, B be two different teams. It follows by the assumption over P that $P'_{A,B} = P_{\pi(A),\pi(B)} > 1/2$, iff $\pi(A) \succ \pi(B)$, which holds iff $A \succ' B$ by definition.

(2) We now show that $\succ'$ is a strict total order. From it's definition we have that $\succ'$ is irreflexive. We also have that $\succ'$ is connected (and therefore strict) as π is invertible and involutory, and every pair of different teams A, B holds either $A \succ' B$ (if $\pi^{-1}(A) = \pi(A) \succ \pi(B) = \pi^{-1}(B)$) or $B \succ' A$ (if $\pi^{-1}(B) = \pi(B) \succ \pi(A) = \pi^{-1}(A)$), but not both. For transitivity, Consider a triplet of different teams, A, B, C such that $A \succ' B \succ' C$ (and therefore $\pi^{-1}(A) = \pi(A) \succ \pi^{-1}(B) = \pi(B) \succ \pi^{-1}(C) = \pi(C)$). From transitivity of $\succ$, we get $\pi^{-1}(A) = \pi(A) \succ \pi(C) = \pi^{-1}(C)$ which implies $A \succ' C$. Hence the relation $\succ'$ is a strict total order.

We continue by showing that $\succ'$ satisfies single players consistency.
Let $x, y \in [n]$ be a pair of players and $S \in [n] \setminus \{x, y\}$ be a set of players such that

$x \cup S \succ' y \cup S$. We will show that $\{x\} \cup S' \succ' \{y\} \cup S'$ for all $S' \in [n] \setminus \{x, y\}$.

Since π is invertible, we know that there exist players $x_\pi = \pi(x)$ and $y_\pi = \pi(y)$, and a set, $S_\pi = \pi(S) \in [n] \setminus \{x_\pi, y_\pi\}$,such that

$$\{x\} \cup S = \pi^{-1}(\{x_\pi\} \cup S_\pi)$$

and

$$\{y\} \cup S = \pi^{-1}(\{y_\pi\} \cup S_\pi).$$

From the definition of $\succ'$, we get

$$\{x_\pi\} \cup S_\pi \succ \{y_\pi\} \cup S_\pi.$$

Therefore from the consistency of $\succ$ every $S'_\pi \in [n] \setminus \{x_\pi, y_\pi\}$ holds $\{x_\pi\} \cup S'_\pi \succ \{y_\pi\} \cup S'_\pi$ hence by definition $\{x\} \cup S' \succ' \{y\} \cup S'$.

(3) We now show that P' satisfy SST w.r.t. $\succ'$. Let $A \succ' B \succ' C$. From the definition of $\succ'$ we have that $\pi^{-1}(A) = \pi(A) \succ \pi^{-1}(B) = \pi(B) \succ \pi^{-1}(C) = \pi(C)$. As P satisfy SST w.r.t. $\succ$,

$$P_{\pi(A),\pi(C)} \geq \max\{P_{\pi(A),\pi(B)}, P_{\pi(B),\pi(C)}\}$$

Once again from the definition of $\succ'$,

$$P'_{A,C} \geq \max\{P'_{A,B}, P'_{B,C}\},$$

Which means that P' satisfy SST w.r.t. $\succ'$ by definition. ∎

Now, observe that, together with the above claim, $a \succ^* b$ imply that for any $S \subseteq [n] \setminus \{a, b\}$ of size $k-1$ it holds that $S \cup \{a\} \succ^* S \cup \{b\}$ which implies (i) $S \cup \{a\} \succ S \cup \{b\}$ as well as (ii) $S \cup \{a\} \succ' S \cup \{b\}$, as both $\succ$ and $\succ'$ are in $\mathcal{C}_{obs}$. Applying the definitions of

$\succ'$ and π, statement (ii) implies $\pi^{-1}(S\cup\{a\}) = \pi(S\cup\{a\}) \succ \pi(S\cup\{b\}) = \pi^{-1}(S\cup\{b\})$ which is equivalent to $S \cup \{b\} \succ S \cup \{a\}$ and hence yields a contradiction to (i).

We therefore deduce the existence of $(A, B) \in \mathcal{D}_a$ such that $P_{A,B} > P_{\pi(A),\pi(B)}$. From (3.3), either $(A, B) \in \mathcal{D}_a^2$, thus $(A\backslash\{a\}, B\backslash\{b\}) \in \mathcal{S}_{a,b}$, and $P_{A,B} > P_{A\backslash\{a\}\cup\{b\},B\backslash\{b\}\cup\{a\}} = P_{\pi(A),\pi(B)}$ yields $(A \setminus \{a\}, B \setminus \{b\}) \in \mathcal{S}^*_{a,b}$, or $(A, B) \in \mathcal{D}_a^3$, thus $(A \setminus \{a\}, B) \in \mathcal{T}_{a,b}$ and $P_{A,B} > P_{A\backslash\{a\}\cup\{b\},B} = P_{\pi(A),\pi(B)}$ implies $(A \setminus \{a\}, B) \in \mathcal{T}^*_{a,b}$ (As $(A, B) \in \mathcal{D}_a^1$, implies $P_{\pi(A),\pi(B)} = P_{A,B} > P_{A,B}$ which is a contradiction.). Overall, $\mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b} \neq \emptyset$. ∎

3.1.7 Algorithms & Proofs for the Stochastic Setting (Section 3.1.4)

We start by splitting the definition of $X_{a,b}(S, S', T)$ into two random variables, according to the two types of witnesses we introduced in the previous section. This will simplify the proof of Lemma 3.4.

For $(S, S') \in \mathcal{S}_{a,b}$ we introduce a random variable $Z_{a,b}(S, S')$ that combines the outcomes of the two duels obtained from the potential subsets witness (S, S'), namely $(S\cup\{a\}, S'\cup\{b\})$ and $(S'\cup\{a\}, S\cup\{b\})$ and similarly, a random variable $Y_{a,b}(S, T)$ that combines the outcomes of the two duels obtained by subset-team witness, $(S \cup \{a\}, T)$ and $(T, S \cup \{b\})$.

Definition 3.22. *For $a, b \in [n], a \neq b$, $(S, S') \in \mathcal{S}_{a,b}$ and $(S, T) \in \mathcal{T}_{a,b}$,*

$$Z_{a,b}(S, S') = \frac{\mathbb{1}[(S \cup \{a\} > S' \cup \{b\})]}{2} + \frac{\mathbb{1}[(S' \cup \{a\} > S \cup \{b\})]}{2},$$
$$Y_{a,b}(S, T) = \frac{\mathbb{1}[\{a\} \cup S > T]}{2} + \frac{\mathbb{1}[T > \{b\} \cup S]}{2}.$$

We note that both $Z_{a,b}(S, S')$ and $Y_{a,b}(S, T)$ can take values in $\{0, 1/2, 1\}$.

The random variables $Z_{a,b}$ and $Y_{a,b}$ are the outcomes of picking random pairs, $(S, S') \in \mathcal{T}_{a,b}$ or $(S, T) \in \mathcal{S}_{a,b}$ and returning $Z_{a,b}(S, S')$ and $Y_{a,b}(S, T)$, respectively. Observe

that

$$\mathbb{E}[Z_{a,b}] = \sum_{(S,S')\in\mathcal{S}_{a,b}} \frac{\mathbb{E}[Z_{a,b}(S,S')]}{|\mathcal{S}_{a,b}|} = \sum_{(S,S')\in\mathcal{S}_{a,b}} \frac{P_{\{a\}\cup S,\{b\}\cup S'} + P_{\{a\}\cup S,\{b\}\cup S'}}{2|\mathcal{S}_{a,b}|},$$
$$\mathbb{E}[Y_{a,b}] = \sum_{(S,T)\in\mathcal{T}_{a,b}} \frac{\mathbb{E}[Y_{a,b}(S,T)]}{|\mathcal{T}_{a,b}|} = \sum_{(S,T)\in\mathcal{T}_{a,b}} \frac{P_{\{a\}\cup S,T} + P_{T,\{b\}\cup S'}}{2|\mathcal{T}_{a,b}|},$$

Where the expectation $\mathbb{E}[Z_{a,b}]$ is taken over all elements of $\mathcal{S}_{a,b}$ and the expectation $\mathbb{E}[Y_{a,b}]$ is taken over all elements $\mathcal{T}_{a,b}$.

The following lemma apply for every $a \succ b$, even if $a \not\succ^* b$. We prove Lemma using SST and consistency.

Lemma 3.23. *Let $a, b \in [n]$ be any two players such that $a \succ b$. Then,*

(1) For every $(S, S') \in \mathcal{S}_{a,b}$ it holds that $\mathbb{E}[Z_{a,b}(S, S')] \geq 1/2$.

(2) For every $(S, T) \in \mathcal{T}_{a,b}$ it holds that $\mathbb{E}[Y_{a,b}(S, T)] \geq 1/2$.

Proof. (1) Let $(S, S') \in \mathcal{S}_{a,b}$ and $a \succ b$. Then,

$$\begin{aligned}
&\mathbb{E}[Z_{a,b}(S,S')] = \frac{P_{\{a\}\cup S,\{b\}\cup S'} + P_{\{a\}\cup S',\{b\}\cup S}}{2} \geq \frac{1}{2} \\
\iff & P_{\{a\}\cup S,\{b\}\cup S'} + P_{\{a\}\cup S',\{b\}\cup S} \geq 1 \\
\iff & P_{\{a\}\cup S,\{b\}\cup S'} \geq 1 - P_{\{a\}\cup S',\{b\}\cup S} \\
\iff & P_{\{a\}\cup S,\{b\}\cup S'} \geq P_{\{b\}\cup S,\{a\}\cup S'},
\end{aligned}$$

which holds according to Lemma 3.21.
(2) Let $(S, T) \in \mathcal{T}_{a,b}$ and $a \succ b$. From Lemma 3.21 we have that

$$P_{\{a\}\cup S,T} \geq P_{\{b\}\cup S,T},$$

which is equivalent to

$$P_{\{a\}\cup S,T} \geq P_{\{b\}\cup S,T} = 1 - P_{T,\{b\}\cup S}$$

and therefore

$$2\mathbb{E}[Y_{a,b}(S,T)] \geq 1.$$

Hence, $\mathbb{E}[Y_{a,b}(S,T)] \geq 1/2$. ∎

Corollary 3.24. *For players $a, b \in [n]$ such that $a \succ b$ then $\mathbb{E}[Z_{a,b}], \mathbb{E}[Y_{a,b}] \geq 1/2$.*

The random variable $X_{a,b}(S, S', T)$ is a single random variable that determines the distinguishability of the relation between players a and b using team duels with potential witnesses. For the definition of $X_{a,b}(S, S', T)$ we refer to the main part of this section.

In the following we show how $X_{a,b}(S, S', T)$ can be expressed by $Z_{a,b}(S, S')$ and $Y_{a,b}(S, T)$, namely

$$\begin{aligned} X_{a,b}(S,S',T) &= \frac{\mathbb{1}[S\cup\{a\} > S'\cup\{b\}] - \mathbb{1}[S\cup\{b\} > S'\cup\{a\}]}{2} \\ &+ \frac{\mathbb{1}[S\cup\{a\} > T] - \mathbb{1}[S\cup\{b\} > T]}{2} \\ &= \frac{\mathbb{1}[S\cup\{a\} > S'\cup\{b\}] + \mathbb{1}[S'\cup\{a\} > S\cup\{b\}] - 1}{2} \\ &\frac{+\mathbb{1}[S\cup\{a\} > T] + \mathbb{1}[T > S\cup\{b\}] - 1}{2} = Z_{a,b}(S,S') + Y_{a,b}(S,T) - 1. \end{aligned}$$

In similar fashion to the definitions of $\mathcal{S}_{a,b}$, $\mathcal{S}^*_{a,b}$ and $Z_{a,b}$ w.r.t. $Z(S, S')$, we defined

$$\mathcal{X}_{a,b} = \{(S, S', T) | (S, S') \in \mathcal{S}_{a,b}, (S, T) \in \mathcal{T}_{a,b}\},$$

and the random variable $X_{a,b}$ to be the outcome of picking a random triplet,

$(S, S', T) \in \mathcal{X}_{a,b}$ and returning $X_{a,b}(S, S', T)$.

The set $\mathcal{X}^*_{a,b}$ contains all triplets $(S, S', T) \in \mathcal{X}_{a,b}$ such that either $(S, S') \in \mathcal{S}^*_{a,b}$ or $(S, T) \in \mathcal{T}^*_{a,b}$. Note that $X_{a,b}(S, S', T), \mathbb{E}[X_{a,b}] \in [-1, 1]$.

For the next Theorem's proof we rely on Theorem 3.2, Corollary 3.24 in one direction, and show the other using the probabilistic method.

Theorem 3.3. *For every two players $a, b \in [n]$ it holds that $a \succ^* b$ if and only if $\mathbb{E}[X_{a,b}] > 0$.*

Proof. We will show that for players $a, b \in [n]$ such that $a \succ^* b$ iff one of the following holds:
(1) $\mathbb{E}[Z_{a,b}] > 1/2$, or
(2) $\mathbb{E}[Y_{a,b}] > 1/2$.
This is equivalent to $\mathbb{E}[X_{a,b}] > 0$ according to the definition of $X_{a,b}$ and Corollary 3.24.
($\Rightarrow$) If $a \succ^* b$ then from Theorem 3.2 we know that one of the following holds:

1. There exists a subsets, witness $(S, S') \in \mathcal{S}_{a,b}$ for $a \succ b$. So by definition $\mathbb{E}[Z_{a,b}(S, S')] > 1/2$, and combined with Lemma 3.23 we have $\mathbb{E}[Z_{a,b}] > 1/2$.
2. There exists a subset-team witness $(S, T) \in \mathcal{T}_{a,b}$ for $a \succ b$. Thus $\mathbb{E}[Y_{a,b}(S, T)] > 1/2$, hence Lemma 3.23 implies that $\mathbb{E}[Y_{a,b}] > 1/2$.

($\Leftarrow$) If (1) holds, the probabilistic method implies the existence of $(S, S') \in \mathcal{S}_{a,b}$ such that $\mathbb{E}[Z_{a,b}(S, S')] > 1/2$ which means that (S, S') is a witness for $a \succ b$, hence, $a \succ^* b$ by Theorem 3.2. If (2) holds, the probabilistic method implies that there exists $(S, T) \in \mathcal{T}_{a,b}$ such that $\mathbb{E}[Y_{a,b}(S, T)] > 1/2$ which means that (S, T) is a witness for $a \succ b$, hence, $a \succ^* b$ by Theorem 3.2.

Thus according to the definition of $X_{a,b}$ the theorem holds. ∎

Gap parameter Recall that we defined our gap parameter by $\Delta = \mathbb{E}[X_{k,k+1}]$. In the following we show that our gap parameter does not just help us to distinguish between the top k and the top $k+1$ players, but also between other players in A_k^* and players from $[n] \setminus A_k^*$. To this end, we show in Lemma 3.4 that strong stochastic transitivity holds for $\mathbb{E}[X_{a,b}]$. For most elements $(S, S', T) \in \mathcal{X}_{a,b}$ it holds that $\mathbb{E}[X_{a,c}(\pi(S), \pi(S'), \pi(T))] \geq \mathbb{E}[X_{a,b}(S, S', T)]$ (and analogously for $X_{b,c}$), where π is a permutation exchanging b and c, but, surprisingly, this is not true in general. By constructing a charging scheme, we can still show that this holds in expectation over all elements of $\mathcal{X}_{a,b}$, and derive a strong stochastic transitivity for distinguishabilities w.r.t. the total order $\succ$ on the players.

The proof of the following lemma also shows that from every $a \succ b$ witness $(S, S', T) \in \mathcal{X}_{a,b}^*$, and for any player c such that $b \succ c$ we can create a $a \succ c$- witness. Similarly, from every $b \succ c$ witness $(S, S', T) \in \mathcal{X}_{b,c}^*$, and for any player a such that $a \succ b$ we can create a $a \succ c$- witness.

Lemma 3.4. *For a triplet of players $a \succ b \succ c$ it holds that*

$$\mathbb{E}[X_{a,c}] \geq \max\{\mathbb{E}[X_{a,b}], \mathbb{E}[X_{b,c}]\}.$$

Proof. In the following we show that $\mathbb{E}[X_{a,c}] \geq \mathbb{E}[X_{a,b}]$. The proof that $\mathbb{E}[X_{a,c}] \geq \mathbb{E}[X_{b,c}]$ works completely analogously and is therefore omitted. Let π be the function exchanging b and c, i.e.

$$\pi(S) = \begin{cases} S \setminus \{c\} \cup \{b\} & \text{if } c \in S, b \notin S \\ S \setminus \{b\} \cup \{c\} & \text{if } b \in S, c \notin S \\ S & \text{else.} \end{cases}$$

Then, we define the function $f : \mathcal{X}_{a,b} \to \mathcal{X}_{a,c}$ by $f(S, S', T) = (\pi(S), \pi(S'), \pi(T))$. Observe that, for this application of π, the second case within the definition of π is never occurs, as none of the sets S, S', T contains b when $(S, S', T) \in \mathcal{X}_{a,b}$. It will we helpful to partition $\mathcal{X}_{a,b}$ in the following way.

$$\begin{aligned}
\mathcal{X}^1_{a,b} &= \{(S, S', T) \in \mathcal{X}_{a,b} \mid c \notin S \cup S' \cup T\} \\
\mathcal{X}^2_{a,b} &= \{(S, S', T) \in \mathcal{X}_{a,b} \mid c \in S\} \\
\mathcal{X}^3_{a,b} &= \{(S, S', T) \in \mathcal{X}_{a,b} \mid c \in S' \setminus T\} \\
\mathcal{X}^4_{a,b} &= \{(S, S', T) \in \mathcal{X}_{a,b} \mid c \in T \setminus S'\} \\
\mathcal{X}^5_{a,b} &= \{(S, S', T) \in \mathcal{X}_{a,b} \mid c \in T \cap S'\}.
\end{aligned}$$

Then we can also define $\mathcal{X}^i_{a,c} = \{f(S, S', T) \mid (S, S', T) \in \mathcal{X}^i_{a,b}\}$ for all $i \in \{1, \ldots, 5\}$. Observe that $\{\mathcal{X}^i_{a,c} \mid i \in \{1, \ldots, 5\}\}$ is also a partition of $\mathcal{X}_{a,c}$.

We will start by proving that for every $(S, S', T) \in \mathcal{X}^1_{a,b} \cup \mathcal{X}^2_{a,b} \cup \mathcal{X}^3_{a,b} \cup \mathcal{X}^4_{a,b} \cup \mathcal{X}^5_{a,b}$

$$\mathbb{E}[Z_{a,c}(f(S, S'))] \geq \mathbb{E}[Z_{a,b}(S, S')] \tag{3.4}$$

and for all $(S, S', T) \in \mathcal{X}^1_{a,b} \cup \mathcal{X}^2_{a,b} \cup \mathcal{X}^3_{a,b}$

$$\mathbb{E}[Y_{a,c}(f(S, T))] \geq \mathbb{E}[Y_{a,b}(S, T)] \tag{3.5}$$

by exhaustion.

(i) Let $(S, S', T) \in \mathcal{X}^1_{a,b}$. We get that $f(S, S', T) = (S, S', T)$ and both

$$\begin{aligned}
\mathbb{E}[Z_{a,c}(S, S')] &= \frac{P_{\{a\}\cup S, \{c\}\cup S'} + P_{\{a\}\cup S', \{c\}\cup S}}{2} \geq \frac{P_{\{a\}\cup S, \{b\}\cup S'} + P_{\{a\}\cup S', \{b\}\cup S}}{2} \\
&= \mathbb{E}[Z_{a,b}(S, S')],
\end{aligned}$$

$$\begin{aligned}\mathbb{E}[Y_{a,c}(S,T)] &= \frac{P_{\{a\}\cup S,T} + P_{T,\{c\}\cup S}}{2} \geq \frac{P_{\{a\}\cup S,T} + P_{T,\{b\}\cup S}}{2} \\ &= \mathbb{E}[Y_{a,b}(S,T)]\end{aligned}$$

follow from consistency and SST.

(ii) Let $(S,S',T) \in \mathcal{X}^2_{a,b}$. Then, $f(S,S',T) = (S \setminus \{c\} \cup \{b\}, S, T)$ and both

$$\begin{aligned}\mathbb{E}[Z_{a,c}(S \setminus \{c\} \cup \{b\}, S')] &= \frac{P_{\{a\}\cup S\setminus\{c\}\cup\{b\},\{c\}\cup S'} + P_{\{a\}\cup S',\{c\}\cup S\setminus\{c\}\cup\{b\}}}{2} \\ &\geq \frac{P_{\{a\}\cup S,\{b\}\cup S'} + P_{\{a\}\cup S',\{b\}\cup S}}{2} \\ &= \mathbb{E}[Z_{a,b}(S,S')]\end{aligned}$$

$$\begin{aligned}\mathbb{E}[Y_{a,c}(S \setminus \{c\} \cup \{b\}, T)]) &= \frac{P_{\{a\}\cup S\setminus\{c\}\cup\{b\},T} + P_{T,\{c\}\cup S}}{2} \\ &\geq \frac{P_{\{a\}\cup S,T} + P_{T,\{b\}\cup S}}{2} \\ &= \mathbb{E}[Y_{a,b}(S,T)]\end{aligned}$$

follow as $\{c\} \cup S \setminus \{c\} \cup \{b\} = S \cup \{b\}$ and from consistency and SST yield the rest.

(iii) Let $(S,S',T) \in \mathcal{X}^3_{a,b}$. Then, $f(S,S',T) = (S, S' \setminus \{c\} \cup \{b\}, T)$ and

$$\begin{aligned}\mathbb{E}[Z_{a,c}(S, S' \setminus \{c\} \cup \{b\}))] &= \frac{P_{\{a\}\cup S,\{c\}\cup S'\setminus\{c\}\cup\{b\}} + P_{\{a\}\cup S'\setminus\{c\}\cup\{b\},\{c\}\cup S}}{2} \\ &\geq \frac{P_{\{a\}\cup S,\{b\}\cup S'} + P_{\{a\}\cup S',\{b\}\cup S}}{2} \\ &= \mathbb{E}[Z_{a,b}(S,S')]\end{aligned}$$

follows as $\{c\} \cup S' \setminus \{c\} \cup \{b\} = S' \cup \{b\}$ and consistency and SST yield the rest. In addition, we already showed that in this case thus $\mathbb{E}[Y_{a,c}(S,T)] \geq \mathbb{E}[Y_{a,b}(S,T)]$

(due to the same reason as in (i)).

(iv) Let $(S, S', T) \in \mathcal{X}^4_{a,b}$. Then, $f(S, S', T) = (S, S', T \setminus \{c\} \cup \{b\})$. Observe that we have already shown that $\mathbb{E}[Z_{a,c}(S, S')] \geq \mathbb{E}[Z_{a,b}(S, S')]$ in this case (due to the same reason as (i)).

(v) Let $(S, S', T) \in \mathcal{X}^5_{a,b}$. Then, $f(S, S', T) = (S, S' \setminus \{c\} \cup \{b\}, T \setminus \{c\} \cup \{b\})$. Observe that we have already shown that $\mathbb{E}[Z_{a,c}(S, S' \setminus \{c\} \cup \{b\})] \geq \mathbb{E}[Z_{a,b}(S, S')]$ in this case (due to the same reason as (iii)).

This concludes the proof of equations (3.4) and (3.5). In particular, from (ii) and (iii) it directly follows that

$$\sum_{(S,S',T) \in \mathcal{X}^i_{a,c}} \mathbb{E}[X(S, S', T)] = \sum_{(S,S',T) \in \mathcal{X}^i_{a,b}} \mathbb{E}[X(f(S, S', T))] \geq \sum_{(S,S',T) \in \mathcal{X}^i_{a,b}} \mathbb{E}[X(S, S', T)] \tag{3.6}$$

holds for $i \in \{2, 3\}$.

We will continue the proof by showing that, for every $(S, T) \in \mathcal{S}_{a,b}$ with $c \in T$, it holds that

$$\mathbb{E}[Z_{a,c}(S, T \setminus \{c\})] + \mathbb{E}[Y_{a,c}(S, T \setminus \{c\} \cup \{b\})] \geq \mathbb{E}[Z_{a,b}(S, T \setminus \{c\})] + \mathbb{E}[Y_{a,b}(S, T)]. \tag{3.7}$$

This will then be helpful to conclude the proof.

To this end, observe that

$$\begin{aligned}
&\mathbb{E}[Z_{a,c}(S, T \setminus \{c\})] + \mathbb{E}[Y_{a,c}(S, T \setminus \{c\} \cup \{b\})]) \\
&= P_{S \cup \{a\}, T} + P_{T \setminus \{c\} \cup \{a\}, S \cup \{c\}} + P_{S \cup \{a\}, T \setminus \{c\} \cup \{b\}} + P_{T \setminus \{c\} \cup \{b\}, S \cup \{c\}} \\
&= P_{S \cup \{a\}, T \setminus \{c\} \cup \{b\}} + P_{T \setminus \{c\} \cup \{a\}, S \cup \{c\}} + P_{S \cup \{a\}, T} + P_{T \setminus \{c\} \cup \{b\}, S \cup \{c\}}
\end{aligned}$$

$$\geq P_{S\cup\{a\},T\setminus\{c\}\cup\{b\}} + P_{T\setminus\{c\}\cup\{a\},S\cup\{b\}} + P_{S\cup\{a\},T} + P_{T,S\cup\{b\}}$$
$$= \mathbb{E}[Z_{a,b}(S, T\setminus\{c\})] + \mathbb{E}[Y_{a,b}(S,T)],$$

which follows by consistency and SST. This will now be helpful to establish a charging scheme. Namely, we are first going to show that

$$\sum_{(S,S',T)\in\mathcal{X}^4_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)] = \sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} \mathbb{E}[X_{a,c}(f(S,S',T))] \geq \sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)]. \tag{3.8}$$

This is true since

$$\sum_{(S,S',T)\in\mathcal{X}^4_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)] + |\mathcal{X}_{a,c}|$$
$$\sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} \mathbb{E}[X_{a,c}(S,S',T\setminus\{c\}\cup\{b\})] + |\mathcal{X}_{a,c}|$$
$$= \sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} (\mathbb{E}[Z_{a,c}(S,S')] + \mathbb{E}[Y_{a,c}(S,T\setminus\{c\}\cup\{b\})])$$
$$= \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,c}(S,T\setminus\{c\}\cup\{b\})]\Big)$$
$$= \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Y_{a,c}(S,S'\cup\{b\})]\Big)$$
$$= \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + \mathbb{E}[Y_{a,c}(S,S'\cup\{b\})]\Big)$$
$$\geq \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \mathbb{E}[Y_{a,b}(S,S'\cup\{c\})]\Big)$$
$$= \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Y_{a,b}(S,S'\cup\{c\})]\Big)$$
$$= \binom{n-k-2}{k-1}\Big(\sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,b}(S,T)]\Big)$$

$$= \sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} (\mathbb{E}[Z_{a,b}(S,S')] + \mathbb{E}[Y_{a,b}(S,T)])$$
$$= \sum_{(S,S',T)\in\mathcal{X}^4_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)] + |\mathcal{X}_{a,b}|,$$

where the inequality follows by equation (3.7). This completes the proof of (3.8).

Next, we are going to show that a similar bound holds when we sum over elements in $\mathcal{X}^1_{a,b} \cup \mathcal{X}^5_{a,b}$. More precisely, we are going to show that

$$\sum_{(S,S',T)\in\mathcal{X}^1_{a,c}\cup\mathcal{X}^5_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)] = \sum_{(S,S',T)\in\mathcal{X}^1_{a,b}\cup\mathcal{X}^5_{a,b}} \mathbb{E}[X_{a,c}(f(S,S',T))]$$
$$\geq \sum_{(S,S',T)\in\mathcal{X}^1_{a,b}\cup\mathcal{X}^5_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)]. \tag{3.9}$$

To this end, observe that

$$\sum_{(S,S',T)\in\mathcal{X}^1_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)] + \sum_{(S,S',T)\in\mathcal{X}^5_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)] + |\mathcal{X}^1_{a,c}| + |\mathcal{X}^5_{a,c}|$$
$$\sum_{(S,S',T)\in\mathcal{X}^1_{a,b}} \mathbb{E}[X_{a,c}(S,S',T)] + \sum_{(S,S',T)\in\mathcal{X}^5_{a,b}} \mathbb{E}[X_{a,c}(S,S'\setminus\{c\}\cup\{b\},T\setminus\{c\}\cup\{b\})] + |\mathcal{X}^1_{a,b}| + |\mathcal{X}^5_{a,b}|$$
$$= \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + \binom{n-k-3}{k-1} \sum_{(S,T)\in\mathcal{T}_{a,b}\cap\mathcal{T}_{a,c}} \mathbb{E}[Y_{a,c}(S,T)]$$
$$+ \binom{n-k-2}{k-1} \sum_{(S,S')\in\mathcal{S}_{a,b}|c\in S'} \mathbb{E}[Z_{a,c}(S,S'\setminus\{c\}\cup\{b\})] + \binom{n-k-2}{k-2} \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,c}(S,T\setminus\{c\}\cup\{b\}$$
$$= \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + [\dots] + \binom{n-k-2}{k-2} \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,c}(S,T\setminus\{c\}\cup\{b\})]$$
$$= \left(\binom{n-k-2}{k} - \binom{n-k-2}{k-2}\right) \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + [\dots]$$
$$+ \binom{n-k-2}{k-2} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,c}(S,S')] + \mathbb{E}[Y_{a,c}(S,S'\cup\{b\})]$$

$$
\begin{aligned}
&\geq \left(\binom{n-k-2}{k} - \binom{n-k-2}{k-2}\right) \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + [\dots] \\
&+ \binom{n-k-2}{k-2} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \mathbb{E}[Y_{a,b}(S,S'\cup\{c\})] \\
&= \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + [\dots] + \binom{n-k-2}{k-2} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Y_{a,b}(S,S'\cup\{c\})] \\
&= \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + [\dots] + \binom{n-k-2}{k-2} \sum_{(S,S')\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,b}(S,T)] \\
&= \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \binom{n-k-3}{k-1} \sum_{(S,T)\in\mathcal{T}_{a,b}\cap\mathcal{T}_{a,c}} \mathbb{E}[Y_{a,c}(S,T)] \\
&+ \binom{n-k-2}{k-1} \sum_{(S,S')\in\mathcal{S}_{a,b}|c\in S'} \mathbb{E}[Z_{a,c}(S,S'\setminus\{c\}\cup\{b\})] + \binom{n-k-2}{k-2} \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,b}(S,T)] \\
&\geq \binom{n-k-2}{k} \sum_{(S,S')\in\mathcal{S}_{a,b}\cap\mathcal{S}_{a,c}} \mathbb{E}[Z_{a,b}(S,S')] + \binom{n-k-3}{k-1} \sum_{(S,T)\in\mathcal{T}_{a,b}\cap\mathcal{T}_{a,c}} \mathbb{E}[Y_{a,b}(S,T)] \\
&+ \binom{n-k-2}{k-1} \sum_{(S,S')\in\mathcal{S}_{a,b}|c\in S'} \mathbb{E}[Z_{a,b}(S,S')] + \binom{n-k-2}{k-2} \sum_{(S,T)\in\mathcal{T}_{a,b}|c\in T} \mathbb{E}[Y_{a,b}(S,T)] \\
&= \sum_{(S,S',T)\in\mathcal{X}^1_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)] + \sum_{(S,S',T)\in\mathcal{X}^5_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)] + |\mathcal{X}^1_{a,c}| + |\mathcal{X}^5_{a,c}|,
\end{aligned}
$$

where the first inequality follows by equation (3.7) and (3.4) and the second inequality follows from equation (3.4) and (3.5). The dots ($[\dots]$) stands for

$$
\binom{n-k-3}{k-1} \sum_{(S,T)\in\mathcal{T}_{a,b}\cap\mathcal{T}_{a,c}} \mathbb{E}[Y_{a,c}(S,T)] + \binom{n-k-2}{k-1} \sum_{(S,S')\in\mathcal{S}_{a,b}|c\in S'} \mathbb{E}[Z_{a,c}(S,S'\setminus\{c\}\cup\{b\})],
$$

which is a part of the expression that it is omitted during the calculations for the

sake of brevity. Summarizing, we get that

$$\begin{aligned}\mathbb{E}[X_{a,c}] &= \frac{\sum_{(S,S',T)\in\mathcal{X}_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)]}{|\mathcal{X}_{a,c}|} = \frac{\sum_{i=1}^{5}\sum_{(S,S',T)\in\mathcal{X}^i_{a,c}} \mathbb{E}[X_{a,c}(S,S',T)]}{|\mathcal{X}_{a,c}|} \\ &\geq \frac{\sum_{i=1}^{5}\sum_{(S,S',T)\in\mathcal{X}^i_{a,b}} \mathbb{E}[X_{a,b}(S,S',T)]}{|\mathcal{X}_{a,c}|} = \mathbb{E}[X_{a,b}],\end{aligned}$$

where the inequality follows from equations (3.6), (3.8), and (3.9). The last inequality follows from $|\mathcal{X}_{a,b}| = |\mathcal{X}_{a,c}|$. ∎

The reduction We close this section by giving the two subroutines mentioned within the reduction to the classic dueling bandits setting.

Algorithm 1 singlesDuel: simulation of a duel between single players

Input: Players $a, b \in [n]$
Output: $w \in \{0, 1/2, 1\}$ such that $w = 1$ is a won, $w = 0$ if b won and $w = 1/2$ in case of a tie.
Pick $(S, S', T) \in \mathcal{X}_{a,b}$ randomly
$z \leftarrow (\mathbb{1}[\{a\} \cup S \succ \{b\} \cup S'] + \mathbb{1}[\{a\} \cup S' \succ \{b\} \cup S])/2$
$y \leftarrow (\mathbb{1}[\{a\} \cup S \succ T] + \mathbb{1}[T \succ \{b\} \cup S])/2$
$x \leftarrow z + y - 1$
return $(1 + x)/2$

Algorithm 2 estimateX: Single player distinguishability estimation

Input: Players $a, b \in [n]$, number of duels $r \in \mathbb{N}$
Output: $x \in [0, 1]$
for $i = 1, \ldots r$ **do**
  $x_i \leftarrow singlesDuel(a, b)$
end for
$x \leftarrow \frac{1}{r}[\sum_{i=1}^{r} x_i] - 1/2$
return x

3.1.8 Algorithms & Proofs for the Deterministic Setting (Section 3.1.5)

Uncover Subroutine As sketched within the main part of this, we refine the idea of the *Uncover* subroutine by a binary search approach. Moreover, we add the option to input a refinement of A and B, namely $A = A^{(1)} \cup A^{(2)}$, $B = B^{(1)} \cup B^{(2)}$, guaranteeing that the uncovered relation is between a pair of players from $A^{(1)}$ and $B^{(1)}$, while $A^{(2)}$ and $B^{(2)}$ are contained in one of the sets of the witness each. For that to work, we require that

(a) $|A^{(1)}| + |A^{(2)}| = k$,

(b) $|A^{(i)}| = |B^{(i)}|$ for $i \in \{1, 2\}$,

(c) $A^{(1)} \cup A^{(2)} \succ B^{(1)} \cup B^{(2)}$, and

(d) $A^{(1)} \cup B^{(2)} \succ B^{(1)} \cup A^{(2)}$.

Observe that for any four sets satisfying (a) and (b) one of the four sets wins in both duels. By enforcing (c) and (d) we fix wlog that this set is $A^{(1)}$. Let us assume that the sets $A^{(1)}$ and $B^{(1)}$ are ordered, meaning that $A^{(1)} = \{a_1, \dots, a_{|A^{(1)}|}\}$ and $B^{(1)} = \{b_1, \dots, b_{|A^{(1)}|}\}$. We also introduce the shorthand notation $A_{\ell:r}$ for $\{a_\ell, \dots, a_r\}$ and respectively $B_{\ell:r}$ for $\{b_\ell, \dots, b_r\}$ for any $\ell, r \in [|A^{(1)}|]$. The subroutine is formalized in Algorithm 3.

Algorithm 3 Uncover Subroutine

Input: four disjoint sets, $A^{(1)}, B^{(1)}, A^{(2)}, B^{(2)}$ with $|A^{(1)}| = |B^{(1)}|, |A^{(2)}| = |B^{(2)}|$, $|A^{(1)}| + |A^{(2)}| = k$, $A^{(1)} \cup A^{(2)} \succ B^{(1)} \cup B^{(2)}$, and $A^{(1)} \cup B^{(2)} \succ B^{(1)} \cup A^{(2)}$

Output: $a \in A^{(1)}, b \in B^{(1)}$, $(S, S') \in \mathcal{S}^*_{a,b}$ with ($C \subseteq S$ and $D \subseteq S'$) or ($D \subseteq S$ and $C \subseteq S'$)

Set $S \leftarrow A^{(1)} \cup A^{(2)}$, $T \leftarrow B^{(1)} \cup B^{(2)}$, $\ell \leftarrow 1$, $r \leftarrow |A^{(1)}|$

while $\ell < r$ **do**

  $i \leftarrow \lfloor \frac{\ell + r}{2} \rfloor$

  $S \leftarrow S - A_{i+1:r} \cup B_{i+1:r}$

  $T \leftarrow T - B_{i+1:r} \cup A_{i+1:r}$

  if $S \succ T$ **then**

    $r \leftarrow i$

  else

    $\ell \leftarrow i + 1$

    swap S and T

  end if

end while

return (a_ℓ, b_ℓ), and $(S \setminus \{a_\ell\}, T \setminus \{b_\ell\})$

In order to show that the algorithm is well-defined and works correctly, the following Lemma will be helpful.

Lemma 3.25. *In subroutine Uncover (Algorithm 3), at the end of every while loop, it holds that, (i) $\ell, r \in \mathbb{N}$ with $\ell \leq r$, (ii) $A_{\ell:r} \subseteq S$, $B_{\ell:r} \subseteq T$, (iii) $S \succ T$, and (iv) $T \setminus B_{\ell:r} \cup A_{\ell:r} \succ S \setminus A_{\ell:r} \cup B_{\ell:r}$, (v) exactly one of S and T contains $A^{(2)}$, the other set contains $B^{(2)}$.*

Proof. We prove all statements via one joint induction over the iterations of the while

loop. All statements are clearly true at the beginning of the first while loop. Now, consider any iteration in which the four statements are true at the beginning of the while loop. It suffices to show that they are still true after resetting S, T, ℓ, and r. For clarity, we refer to the modified variables of the teams just before the if condition as S', T' and after the if condition as S'', T''. Similarly, ℓ', and r' are the values of the indices after the if condition. In the following, we show that the four conditions still hold for S'',T'',ℓ', and r'.

Case 1: $S' \succ T'$. Then, $S'' = S'$, $T'' = T'$, $\ell' = \ell$, $r' = i$. The condition of the while loop, $\ell < r$, clearly implies that $\ell' = \ell \leq \lfloor \frac{\ell+r}{2} \rfloor = i = r'$. Moreover, by construction $A_{\ell:i} = A_{\ell':r'} \subseteq S''$ and $B_{\ell:i} = B_{\ell':r'} \subseteq T''$ and hence condition (ii) is satisfied. Condition (iii), i.e., $S'' \succ T''$ is satisfied by the case condition. For condition (iv) let us rewrite the induction hypothesis for condition (iv) as

$$T - B_{\ell:r} \cup (A_{\ell:i} \cup A_{i+1:r}) \succ S - A_{\ell:r} \cup (B_{\ell:i} \cup B_{i+1:r}).$$

Observe that $T - B_{\ell:r} \cup A_{i+1:r} = T' - B_{\ell:i}$ and $S - A_{\ell:r} \cup B_{i+1:r} = S' - A_{\ell:i}$. Hence, the above expression can be rewritten as

$$T' - B_{\ell:i} \cup A_{\ell:i} \succ S' - A_{\ell:i} \cup B_{\ell:i}.$$

Plugging in $T' = T''$, $S' = S''$, $\ell = \ell'$ and $i = r'$ yields condition (iv) for the updated variables. Lastly, condition (v) is satisfied directly by applying the induction hypothesis.

Case 2: $T' \succ S'$. Then, $S'' = T'$, $T'' = S'$, $\ell' = i + 1$, $r' = r$. For condition (ii), observe that $\ell, r \in \mathbb{N}$ with $\ell < r$ clearly implies that $\ell' = i + 1 = \lfloor \frac{\ell+r}{2} \rfloor + 1 \leq \lfloor \frac{2r-1}{2} \rfloor + 1 \leq r = r'$. Moreover, by construction $A_{i+1:r} \subseteq T' = S''$ and $B_{i+1:r} \subseteq S' = T''$ and hence (ii) is satisfied. Condition (iii), i.e., $S'' = T' \succ S' = T''$, is satisfied by

the case condition. For condition (iv), let us rewrite the induction hypothesis for condition (iii) as

$$S - A_{\ell:r} \cup (A_{\ell:i} \cup A_{i+1:r}) \succ T - B_{\ell:r} \cup (B_{\ell:i} \cup B_{i+1:r}).$$

Observe that $S - A_{\ell:r} \cup A_{\ell:i} = S' - B_{i+1:r}$ and $T - B_{\ell:r} \cup B_{\ell:i} = T' - A_{i+1:r}$. Hence, the above expression can be rewritten to

$$S' - B_{i+1:r} \cup A_{i+1:r} \succ T' - A_{i+1:r} \cup B_{i+1:r}.$$

Inserting $S' = T''$, $T' = S''$, $i + 1 = \ell'$ and $r = r'$ yields condition (iv) for the updated variables. Lastly, condition (v) is satisfied directly by applying the induction hypothesis. ∎

With the help of Lemma 3.25 it is easy to see that the algorithm is well-defined, more precisely, that the constructed tuple (S, T) forms a feasible duel within every iteration of the while loop. It remains to show that the algorithm works correctly and its running time is bounded by $\mathcal{O}(log(|A^{(1)}|))$.

Lemma 3.26. *Let $A^{(1)}, A^{(2)}, B^{(1)}, B^{(2)}$ be sets satisfying conditions (a) to (d). After performing $\mathcal{O}(\log(|A^{(1)}|))$ duels,* Uncover *returns (a, b) with $a \in A^{(1)}$, $b \in B^{(1)}$ and $(S, S') \in \mathcal{S}^*_{a,b}$ with either $A^{(2)} \subseteq S$ and $B^{(2)} \subseteq S'$ or $B^{(2)} \subseteq S$ and $A^{(2)} \subseteq S'$.*

Proof. By Lemma 3.25, the termination of the algorithm implies that $\ell = r$. By statement (ii) from Lemma 3.25 we get that $a_\ell \in S$ and $b_\ell \in T$ holds. Moreover, conditions (iii) and (iv) can be rewritten as

$$(S \setminus \{a_\ell\}) \cup \{a_\ell\} \succ (T \setminus \{b_\ell\}) \cup \{b_\ell\} \tag{3.10}$$

and

$$(T \setminus \{b_\ell\}) \cup \{a_\ell\} \succ (T \setminus \{a_\ell\}) \cup \{b_\ell\}, \tag{3.11}$$

respectively. Clearly, this implies that $(S \setminus \{a_\ell\}, T \setminus \{b_\ell\}) \in \mathcal{S}^*_{a_\ell,b_\ell}$ and hence $a_\ell \succ b_\ell$.

It is easy to see that the number of iterations of the while loop is upper bounded by the height of a balanced binary tree on $|A^{(1)}|$ elements, i.e., $\mathcal{O}(log(|A^{(1)}|))$. Since every iteration induces exactly one query, this also bounds the total number of queries. Moreover, by condition (v) we have that one of $A^{(2)}$ is included in S or T and $B^{(2)}$ in the other one. This concludes the proof. ∎

Clearly, Lemma 3.26 directly implies Lemma 3.10. For this, simply call *Uncover* with $A^{(2)} = B^{(2)} = \emptyset$.

Lemma 3.10. *Let A and B be two disjoint teams with $A \succ B$. After performing $\mathcal{O}(\log(k))$ duels,* Uncover *returns (a, b) with $a \in A$, $b \in B$ and $(S, S') \in \mathcal{S}^*_{a,b}$.*

Reducing the Number of Players to $\mathcal{O}(k)$ Before formalizing the pre-processing procedure *ReducePlayers* in Algorithm 4, recall that algorithm maintains a dominance graph $D = (V, E)$ on the set of players. More precisely, the nodes of D are the players, i.e., $V = [n]$, and there exists an arc from node a to node b if the algorithm has proven that $a \succ b$. The set $V_{<2k}$ is the subset of the players having an indegree smaller than $2k$ in D.

Additionally, we define a second graph $G_{<2k}$ as follows: The set of nodes of $G_{<2k}$ equals $V_{<2k}$ and there exists an (undirected) edge between two nodes $a, b \in V_{<2k}$ if and only if neither of the arcs (a, b) or (b, a) is present within the graph D. The algorithm now searches for a matching of size k within the graph $G_{<2k}$ by calling the subroutine *GreedyMatching*, formalized in Algorithm 5. Let $\{(a_1, b_1), \ldots, (a_k, b_k)\}$ be such a matching. In particular, this implies that the algorithm has not identified any

of the relations between a_i and b_i yet. Hence, when calling *uncover* for the (ordered) sets $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$ (after possibly swapping A and B), the algorithm learns about one additional pairwise relation, say $a_i \succ b_i$ and add the arc (a_i, b_i) to the graph D. Then, the algorithm also updates D to its transitive closure. The algorithm ends when it cannot find a matching of size k in $G_{<2k}$ anymore. We formalize the idea within Algorithm 4.

Algorithm 4 ReducePlayers

Input: a set of players $[n]$

Output: a set S with $|S| \leq 6k - 2$ s.t. $A^*_{2k} \subseteq S$

while $|\mathit{GreedyMatching}(G_{<2k})| = k$ **do**

Let $\{\{a_1, b_1\}, \ldots, \{a_k, b_k\}\}$ be Greedy Matching

Set $A = \{a_1, \ldots, a_k\}, B = \{b_1, \ldots, b_k\}$

$(a, b) \leftarrow \mathit{uncover}(A, B)$

Add (a, b) to D, $D \leftarrow \mathit{transitiveClosure}(D)$

Update $V_{<2k}$ and $G_{<2k}$

end while

return $V_{<2k}$

Algorithm 5 Subroutine GreedyMatching

Input: an undirected Graph $G = (V, E)$

Output: a matching of size at most k

$M \leftarrow \emptyset$

while $|M| < k$ and $E \neq \emptyset$ **do**

Pick arbitrary edge (u, v) from E

Delete all edges incident to u and v from E

end while

return M

Lemma 3.12. *Given the set of players* $[n]$*,* ReducePlayers *returns* $X \subseteq [n]$ *with* $|X| \leq 6k - 2$ *and* $A^*_{2k} \subseteq X$*.* ReducePlayers *performs* $\mathcal{O}(nk \log(k))$ *duels and runs in time* $\mathcal{O}(n^2k^2)$*.*

Proof. We start by proving that $A^*_{2k} \subseteq X$. Every player not included in X has at least $2k$ ingoing arcs in D. In other words, there exist $2k$ players which dominate it. Hence, such a player is not included in A^*_{2k}.

We turn to prove that $|X| \leq 6k-2$: Any *independent set* within the graph $G_{<2k}$ contains less than $2k+1$ nodes. An independent set within $G_{<2k}$ is a subset of the nodes $T \subseteq V_{<2k}$ such that no two nodes of T are connected by an edge. Now, assume for contradiction that there exists an independent set $T \subseteq V_{<2k}$ within the graph $G_{<2k}$ with $|T| = 2k+1$. Consider the subgraph of D induced by the set T, i.e., $D[T] = (T, \{(a,b) \in E \mid a,b \in T\})$. Since T is an independent set within $G_{<2k}$, we know that $D[T]$ is a tournament graph, i.e., a directed graph in which any two nodes are connected by exactly one directed arc. Moreover, since $D[T]$ is transitive (since $\succ$ and hence D is transitive), there exists exactly one node within T with an indegree of $2k$ within the graph D. This is a contradiction to $T \subseteq V_{<2k}$.

This observation is now helpful to conclude the proof. Assume for contradiction that $|V_{<2k}| \geq 6k-1$. Then the following greedy procedure lets us construct a matching of size $2k$ within the graph $G_{<2k}$. This yields a contradiction to the termination of the while loop, since every maximal matching, and in particular, a matching of size smaller than k returned by *GreedyMatching*, is a 1/2-approximation of a matching with maximum cardinality. Hence, the existence of a matching with $2k$ edges yields a contradiction to the fact that *GreedyMatching* did not find a matching of size k. We start by defining $T = V_{<2k}$ and $M = \emptyset$. Since $|T| > 2k$, T is not an independent set and there exists an edge between some two nodes in T. Now, pick any such edge, say $\{a,b\}$, and add it to M and remove a and b from T. After i rounds of this procedure, $|M| = i$ and $|T| = 2k + 2(2k-i) - 1$. We can repeat this procedure for $2k$ rounds and have found a matching of size $2k$, a contradiction.

We now turn to prove the number of duels performed by the algorithm. In every step of the while loop, the algorithm adds one arc which was not existent before to the graph D. Moreover, since any selected matching never includes an edge with one of its endpoints having an indegree larger than $2k-1$, no node has an indegree

higher than $2k$ after the termination of the algorithm. We can then upper bound the number of arcs within D by $2kn$.

This is also a bound for the number of iterations of the while loop. Within each iteration of the while loop the algorithm needs to make one query in order to identify the winning team and in addition it calls the subroutine *uncover*. As argued within the proof of Lemma 3.12, the *uncover* subroutine induces additional $\mathcal{O}(log(k))$ queries per while loop. Summarizing, this implies that the algorithm requires $\mathcal{O}(nk \log(k))$ queries in total.

As for the running time, we have already argued that the while loop does at most $\mathcal{O}(nk)$ iterations. Within the while loop the algorithm needs to run *GreedyMatching* for finding a matching of size k within $G_{<2k}$ and run the *uncover* subroutine. While the latter step requires a running time of $\mathcal{O}(\log(k))$ as argued within Lemma 3.12, *GreedyMatching* for selecting a matching of size k can be implemented in $\mathcal{O}(nk)$. In total, we get a running time of $\mathcal{O}(n^2k^2)$. ∎

Subroutines NewCut and Compare In Algorithm 6 we formalize the subroutine *NewCut*, which takes as input a subset of the players $X \subseteq [n]$, a pair of players $a, b \in X$ and a witness $(S, T) \in \mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$ and outputs a partition of X into U and L such that $U \triangleright L$ holds. We denote by π_{xy} the permutation on subsets that exchange players x and y. More precisely,

$$\pi_{xy}(A) = \begin{cases} A \setminus \{x\} \cup \{y\} & \text{if } x \in A, y \notin A \\ A \setminus \{y\} \cup \{x\} & \text{if } x \notin A, y \in A \\ A & \text{else.} \end{cases}$$

Algorithm 6 NewCut

Input: $X \subseteq [n]$, a pair $a, b \in X$ and $(S, T) \in \mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$

Output: Partition of X into $U \triangleright L$ with $a \in U$ and $b \in L$

Initialize $\mathcal{W} \leftarrow \{(S, T, a)\}$, $U \leftarrow \{a\}, X \leftarrow X \setminus \{a, b\}$

while $\mathcal{W}$ non-empty **do**

  Pick $(S, T, y) \in \mathcal{W}$ and remove it from $\mathcal{W}$

  for $x \in X$ **do**

    if $(\pi_{xy}(S), \pi_{xy}(T)) \in \mathcal{S}^*_{xb} \cup \mathcal{T}^*_{xb}$ **then**

      add x to U, remove x from X

      add $(\pi_{xy}(S), \pi_{xy}(T), x)$ to $\mathcal{W}$

    else if $|T| = k$ and $x \in T$ and $(S, T \setminus \{x\}) \in \mathcal{S}^*_{xb}$ **then**

      add x to U and remove it from X

      add $(S, T \setminus \{x\}, x)$ to $\mathcal{W}$

    end if

  end for

end while

return $(U, X \cup \{b\})$

Before we prove the correctness of the algorithm, we introduce the following two lemmas. Strictly speaking, these are special cases of statements shown within the proof of Lemma 3.4 for the deterministic setting. For the sake of illustration, we state and prove them here for the deterministic case again, independently of Lemma 3.4.

Lemma 3.27. *If $a \succ b \succ c$ and $(S, S') \in \mathcal{S}^*_{b,c}$, then $(\pi_{ab}(S), \pi_{ab}(S')) \in \mathcal{S}^*_{a,c}$.*

Proof. We distinguish two cases. First assume $a \notin S \cup S'$. Then,

$$S \cup \{a\} \succ S \cup \{b\} \succ S' \cup \{c\},$$

where the first statement follows from single-player consistency and the second statement from $(S, S') \in \mathcal{S}^*_{bc}$. Moreover,

$$S' \cup \{a\} \succ S' \cup \{b\} \succ S \cup \{c\},$$

where again the first statement follows from single-player consistency and the second one from $(S, S') \in \mathcal{S}^*_{bc}$.

If $a \in S \cup S'$, assume wlog that $a \in S$. Then, $\pi_{ab}(S) = S \setminus \{a\} \cup \{b\}$ and $\pi_{ab}(S') = S'$. We get

$$\pi_{ab}(S) \cup \{a\} = S \cup \{b\} \succ S' \cup \{c\}$$

and

$$S' \cup \{a\} \succ S' \cup \{b\} \succ S \cup \{c\} = S \setminus \{a\} \cup \{a\} \cup \{c\} \succ S \setminus \{a\} \cup \{b\} \cup \{c\} = \pi_{ab}(S) \cup \{c\},$$

where the first and last statement follow from single player consistency and the second statement from $(S, S') \in \mathcal{S}^*_{bc}$. Summarizing, $(\pi_{ab}(S), \pi_{ab}(S')) \in \mathcal{S}^*_{ac}$. ∎

Lemma 3.28. *If $a \succ b \succ c$ and $(S, T) \in \mathcal{T}^*_{b,c}$, then $(\pi_{ab}(S), \pi_{ab}(T)) \in \mathcal{T}^*_{a,c}$ or $(S, T \setminus \{a\}) \in \mathcal{S}^*_{a,c}$.*

Proof. We distinguish three cases. First, assume that $a \notin S \cup T$. Then,

$$S \cup \{a\} \succ S \cup \{b\} \succ T \succ S \cup \{c\},$$

where the first statement follows from single player consistency and the second and third from $(S,T) \in \mathcal{T}^*_{bc}$. Next, assume $a \in S$. Then, $\pi_{ab}(S) = S \setminus \{a\} \cup \{b\}$ and we get

$$\pi_{ab}(S) \cup \{a\} = S \cup \{b\} \succ T \succ S \cup \{c\} = S \setminus \{a\} \cup \{a\} \cup \{c\} \succ S \setminus \{a\} \cup \{b\} \cup \{c\} = \pi_{ab}(S) \cup \{c\}.$$

Hence, $(\pi_{ab}(S), \pi_{ab}(T)) \in \mathcal{T}^*_{ab}$. Finally, assume $a \in T$. We get,

$$S \cup \{a\} \succ S \cup \{b\} \succ T \setminus \{a\} \cup \{a\} \succ T \setminus \{a\} \cup \{c\},$$

where the first and last statement follow from single player consistency and the second statement from $(S,T) \in \mathcal{T}^*_{bc}$. Moreover,

$$T \setminus \{a\} \cup \{a\} \succ S \cup \{c\},$$

which follows from $(S,T) \in \mathcal{T}^*_{bc}$. Summarizing, $(S, T \setminus \{a\}) \in \mathcal{S}^*_{ac}$. ∎

Having these two lemmas, we are ready to prove the correctness of the *NewCut* subroutine.

Lemma 3.13. *Let $X \subseteq [n]$, $a, b \in X$ and $(S,T) \in \mathcal{S}^*_{a,b} \cup \mathcal{T}^*_{a,b}$. Then,* NewCut$(X, (a,b), (S,T))$ *returns a partition of X into U and L such that $U \triangleright L$, $a \in U$ and $b \in L$. The number of duels performed by* NewCut *and its running time can be bounded by $\mathcal{O}(|X|^2)$.*

Proof. Let X be the original set of players given as input to the algorithm, and U and L the returned sets. We denote by X' and U' the corresponding sets maintained and modified by the algorithm during its execution. To see that U and L form a partition of V, observe that U' and X' form a partition of $X \setminus \{b\}$ during the entire execution of the algorithm.

We turn to show that $U \triangleright L$. Assume for contradiction that there exists $c \in L$ and $d \in U$ with $c \succ d$. Since $d \in U$ we know that the algorithm found a witness for $d \succ b$ which we denote by (S, T) and added (S, T, d) to the list $\mathcal{W}$. Moreover, as $c \in L$, the algorithm selected $x = c$ in the for loop when (S, T, d) was picked from $\mathcal{W}$. Now, if $|T| = k - 1$, we know that $(S, T) \in \mathcal{S}^*_{d,b}$ and can apply Lemma 3.27 which yields $(\pi_{cd}(S), \pi_{cd}(T)) \in S^*_{c,b}$. This is a contradiction, as otherwise c would have been added to U' at this point. If $|T| = k$, we can apply Lemma 3.28, yielding that either $(\pi_{cd}(S), \pi_{cd}(T)) \in \mathcal{T}_{c,b}$ or $(S, T \setminus \{c\}) \in \mathcal{S}^*_{c,b}$, both of which cannot be as $c \notin U'$ at the end of the algorithm. This completes the proof of correctness.

It remains to bound the number of duels performed. Since the number of duels performed in every iteration of the for loop is constant, it suffices to bound the number of iterations of the for loop. As the algorithm adds at most $|X| - 1$ elements to $\mathcal{W}$ and for each element the for loop runs at most $|X| - 2$ times, the number of duels can be bounded by $\mathcal{O}(|X|^2)$. ∎

We now turn to formalize the subroutine *Compare* within Algorithm 7.

Algorithm 7 Compare

Input: tuple (a, b), witness $(S, S') \in \mathcal{S}^*_{ab}$ and $C \subseteq S$, $D \subseteq S'$ with $|C| = |D|$

if $S \setminus C \cup D \cup \{a\} \succ S' \setminus D \cup C \cup \{b\}$ and $S' \setminus D \cup C \cup \{a\} \succ S \setminus C \cup D \cup \{b\}$ **then**

  return True

else

  return False

end if

Lemma 3.14. *Let $a \succ b$ be two players, $(S, S') \in \mathcal{S}^*_{a,b}$ and $C \subseteq S, D \subseteq S'$ with $|C| = |D|$. If* Compare$((a, b), (S, S'), (C, D))$ *returns* True*, then $v(a) - v(b) > |v(C) - v(D)|$. Otherwise, one call to* Uncover *returns $c \in C$ and $d \in D$ together with a witness for*

their relation.

Proof. For the sake of brevity we define $\bar{S} = S \setminus C$ and $\bar{S}' = S \setminus D$. Recall that from $(S, S') \in \mathcal{S}_{a,b}$ we get that (i) $\bar{S} \cup C \cup \{a\} \succ \bar{S}' \cup D \cup \{b\}$ and (ii) $\bar{S}' \cup D \cup \{a\} \succ \bar{S} \cup C \cup \{b\}$ hold. Recall that we are considering additive total orders. For any set $A \subseteq [n]$ we define $v(A) = \sum_{a \in A} v(a)$. Then, we can rewrite (i) and (ii) to

$$(i)\ v(\bar{S}) + v(C) + v(a) > v(\bar{S}') + v(D) + v(b)$$

and

$$(ii)\ v(\bar{S}') + v(D) + v(a) > v(\bar{S}') + v(C) + v(b).$$

Then, we distinguish two cases.

Case 1. (iii) $\bar{S} \cup D \cup \{a\} \succ \bar{S}' \cup C \cup \{b\}$ and (iv) $\bar{S}' \cup C \cup \{a\} \succ \bar{S} \cup D \cup \{b\}$. Similarly to before, we can rewrite (iii) and (iv) to

$$(iii)\ v(\bar{S}) + v(D) + v(a) > v(\bar{S}') + v(C) + v(b)$$

and

$$(iv)\ v(\bar{S}') + v(C) + v(a) > v(\bar{S}) + v(D) + v(b).$$

Then, from adding (ii) and (iii) we get that

$$v(a) - v(b) > v(C) - v(D)$$

and from adding (i) and (iv) we get that

$$v(a) - v(b) > v(D) - v(C).$$

Summarizing, this yields $v(a) - v(b) > |v(C)| - |v(D)|$.

Case 2. (v) $\bar{S}' \cup C \cup \{b\} \succ \bar{S} \cup D \cup \{a\}$

In that case, observe that the quartet $(C, D, \bar{S} \cup \{a\}, \bar{S}' \cup \{b\})$ satisfies the requirements for the *Uncover* subroutine due to equation (i) and (v). Hence, *Uncover* will return a dominance of some player in C towards some player in D together with a witness for this relationship.

Case 3. (vi) $\bar{S} \cup D \cup \{b\} \succ \bar{S}' \cup C \cup \{a\}$

In that case, observe that the quartet $(D, C, \bar{S} \cup \{b\}, \bar{S}' \cup \{a\})$ satisfies the requirements for the *Uncover* subroutine due to equation (ii) and (vi). Hence, *Uncover* will return a dominance of some player in D towards some player in C together with a witness for this relationship. ∎

Algorithm CondorcetWinning Recall that the algorithm maintains a partition of the players into a weak ordering, i.e., $\mathcal{T} = \{T_1, \dots, T_\ell\}$ with $T_1 \triangleright T_2 \triangleright \cdots \triangleright T_\ell$. We introduce the short-hand notation $T_{\leq j} = \bigcup_{m \in [j]} T_m$ and $T_{<j} = \bigcup_{m \in [j-1]} T_m$. After the application of the preprocessing procedure *ReducePlayers*, this partition consists of one set, namely $\mathcal{T} = \{T_1\}$, where $|T_1| \in \mathcal{O}(k)$ and $A^*_{2k} \subseteq T_1$. At any point in the execution of the algorithm, we are especially interested in two indices, namely $i_k \in [\ell]$ such that $|T_{<i_k}| < k < |T_{\leq i_k}|$ and similarly $i_{2k} \in [\ell]$ such that $|T_{<i_{2k}}| < 2k < |T_{\leq i_{2k}}|$. In case one of these indices does not exist, this implies that we have either identified the set A^*_k or A^*_{2k}. In the first case, we have found a Condorcet winning team and in the second case Observation 3.11 implies that we can find one by performing one additional duel. For the sake of brevity, we disregard this case from now on.

Assuming i_k is defined, observe that all players from $T_{<i_k}$ are guaranteed to be among the top-k players. On the other hand, among the players from T_{i_k} some belong to A^*_k

and others do not. The main idea of the algorithm will then be to, at any given time, take some k-sized prefix of $\mathcal{T}$, i.e., a subset including $T_{<i_k}$ that is included in $T_{\leq i_k}$ and either proving that this prefix is a Condorcet winning team, or showing that the partition $\mathcal{T}$ can be refined.

In the following we distinguish the cases that $i_k \neq i_{2k}$ and $i_k = i_{2k}$. For the first case we give the algorithm *CondorcetWinning1* and for the latter case the algorithm *CondorcetWinning2*. Observe that, once the *CondorcetWinning1* called *CondorcetWinning2* (which implies $i_k \neq i_{2k}$) this will be true until the termination of the algorithm.

CondorcetWinning1 The algorithm starts by partitioning the set $T_{<i_k}$ into two sets U_1 and U_2, where U_1 is a prefix of $T_{<i_k}$ of size $|T_{\leq i_k}| - 2k$. It partitions the set T_{i_k} into five sets X, Y, W_1, W_2, and Z. In particular it is known that $(U_1 \cup U_2) \triangleright (X \cup Y \cup W_1 \cup W_2 \cup Z)$ but no relation among any pair in T_{i_k} is known. Regarding the sizes of the sets it holds that $|U_i| = |W_i|$ for $i \in \{1, 2\}$, $|X| = |Y| = k - |U_1| - |U_2|$ and $|U_1| = |Z|$. The main aim of the algorithm will be to define $0 < \epsilon_1 < \epsilon_2$ and prove that the following statements are true:

(i) $|v(X) - v(Y)| < \epsilon_1$

(ii) $|v(a) - v(b)| < \epsilon_2$ for all $a \in Y \cup W_1 \cup W_2$ and $b \in Z$, and

(iii) there exist $u_1, \ldots, u_{|Z|+1} \in U_1 \cup U_2$ as well as $w_1, \ldots, w_{|Z|+1} \in W_1 \cup W_2$ such that

  (a) $v(u_1) - v(w_1) \geq \epsilon_1$ and

  (b) $v(u_i) - v(w_i) \geq \epsilon_2$ for all $i \in \{2, \ldots, |Z| + 1\}$.

With these three statements we can show that $U_1 \cup U_2 \cup X$ is a Condorcet winning

team. More precisely, one can show that $v(U_1 \cup U_2 \cup X) - v(W_1 \cup W_2 \cup Y) > |Z| \cdot \epsilon_2$ and $v(W_1 \cup W_2 \cup Y) - v(B^*) > -|Z| \cdot \epsilon_2$, where B^* is the best response towards $U_1 \cup U_2 \cup X$, i.e., B^* simply contains the best k players from $[n] \setminus (U_1 \cup U_2 \cup X)$. See Figure 3.1 for an illustration of the argument.

It remains to sketch how the algorithm defines ϵ_1, ϵ_2 and proves $(i) - (iii)$. The algorithm starts by checking whether *Uncover* can be applied to the sets $A^{(1)} = U_2, A^{(2)} = X \cup Z, B^{(1)} = W_2, B^{(2)} = Y \cup W_1$. If this is not the case, a relation between a pair in $A^{(2)}$ and $B^{(2)}$ can be found and the partition can be refined by applying *NewCut*. Otherwise, let $\bar{u} \in U_2$ and $\bar{w} \in W_2$ be the returned pair from *Uncover*. For the sake of brevity we assume for now that the entire indifference class of $\bar{u}$ in $\mathcal{T}$ is included in U_2. Then, using *Compare*, the algorithm checks whether $|v(X) - v(Y)| < v(\bar{u}) - v(\bar{w})$ and whether $|v(a) - v(b)| < v(\bar{u}) - v(\bar{w})$ for all $a \in W_1 \cup W_2 \cup Y$ and $b \in Z$. The algorithm repeats the process by replacing $\bar{w}$ by all $w \in W_1$. If any of the calls to *Compare* returned *False*, then we show that the partition can be refined. Otherwise, we have shown that conditions $(i) - (iii)$ are satisfied for $\epsilon_1 = v(\bar{u}) - v(w_1^*)$ and $\epsilon_2 = v(\bar{u}) - v(w_2^*)$, where w_1^* and w_2^* are the best and second best players from $W_1 \cup \{\bar{w}\}$, respectively. For the case when not the entire indifference class of $\bar{u}$ is included in U_2, we still have to exchange $\bar{u}$ by other players from its indifferent class which are included in U_1.

Lemma 3.29. *After performing $\mathcal{O}(k^5)$ many duels,* CondorcetWinning1 *has identified a Condorcet winning team or called* CondorcetWinning2.

Proof. In part I we show that the algorithm is well-defined and that, within line 13,21,24, 29, 35, and 40, a refined partition can indeed be found. In part II we show that, if the algorithm outputs a team, this team is indeed Condorcet winning. Lastly, in part III we argue about the bound on the number of duels performed.

Part I. We show the first two statements by going through the algorithm line by line.

We start by showing that in line 12, the two queries are feasible. First observe that by construction, the sets U_1, U_2, X, Y, W_1, W_2, and Z are disjoint. Moreover, $|U| = |W|$, $|U_1| = |W_1|$, and hence $|U_2| = |W_2|$. Also, $|X| = |Y|$ and $|W_1| = |Z|$. In total, we get that $|W_2| + |Y| + |W_1| = |U_1| + |X| + |Z| = |U| + |X| = k$ and the same holds for the other query as well.

Next, we show that in line 13, the partition $\mathcal{T}$ can indeed be refined. Consider wlog the case when $W_2 \cup (Y \cup W_1) \succ U_2 \cup (X \cup Z)$. Then, since $U_2 \triangleright W_2$ we know that $U_2 \cup (Y \cup W_1) \succ W_2 \cup (X \cup Z)$ needs to hold. Hence, Uncover$(Y \cup W_1, X \cup Z, W_2, U_2)$ returns a pair (a, b) with $a \in Y \cup W_1$ and $b \in X \cup Z$ together with a witness $(S, S') \in \mathcal{S}_{a,b}$. Since $a, b \in T_{i_k}$, we can call NewCut$(\mathcal{T}, (a, b), (S, S'))$ which returns a refined partition. An analogous argument holds for the case $W_2 \cup (X \cup Z) \succ U_2 \cup (Y \cup W_2)$.

We turn to show that the input for the Uncover subroutine is valid in line 15. Since the condition in line 12 is not satisfied, we know that $U_2 \cup (X \cup Z) \succ W_2 \cup (Y \cup W_1)$ and $U_2 \cup (Y \cup W_1) \succ W_2 \cup (X \cup Z)$. This suffices to show that $(U_2, W_2, (X \cup Z), (Y \cup W_1))$ is a valid input for Uncover. Hence, for the returned pair $(\bar{u}, \bar{w})$ is holds that $\bar{u} \in U_2$ and $\bar{w} \in W_2$. Moreover, we can assume in the following wlog that $(X \cup Z) \subseteq S$ and $(Y \cup W_1) \subseteq S'$.

We continue with the situation in line 21 and show that a refined partition can be found. We distinguish two cases.

Case 1 $(S, S'') \in \mathcal{S}_{\bar{u},w}$. This implies (i) $S \cup \{\bar{u}\} \succ S'' \cup \{w\}$ and (ii) $S'' \cup \{\bar{u}\} \succ S \cup \{w\}$. Moreover, from $(S, S'') \notin \mathcal{S}_{u,w}$ we know that either (iii) $S \cup \{w\} \succ S'' \cup \{u\}$ or (iv) $S'' \cup \{w\} \succ S \cup \{u\}$ is true. Assume without loss of generality that (iii) holds. Then, together with (ii) we get that $S'' \cup \{\bar{u}\} \succ S \cup \{w\} \succ S'' \cup \{u\}$, hence $\bar{u} \succ u$ and in particular $(S \cup \{w\}, S'') \in \mathcal{T}_{\bar{u},u}$. Since $\bar{u}$ and u are from the same indifference

class of $\mathcal{T}$, calling NewCut2($\mathcal{T}, (\bar{u}, u), (S \cup \{w\}, S'')$) returns a refined partition. An analogous argument holds when (iv) is true.

Algorithm 8 CordorcetWinning1

Input: a partition of $[n]$ into $T_1 \triangleright T_2 \triangleright \cdots \triangleright T_\ell$

Output: a CondorcetWinning Team

if $i_k \neq i_{2k}$ **then**
  return CondorcetWinning2($\mathcal{T}$)
end if
Set $U \leftarrow T_{<i_k}$
Set X and Y to be two disjoint, $(k - |U|)$-sized subsets of T_{i_k}
Set W to be a $|U|$-sized subset of $T_{i_k} \setminus X \setminus Y$
Set Z to be $T_{i_k} \setminus X \setminus Y \setminus W$
Set W_1 to be a $|Z|$-sized subset of W and $W_2 \leftarrow W \setminus W_1$
Set U_1 to be a $|Z|$-sized prefix of U and $U_2 \leftarrow U \setminus U_1$
if $W_2 \cup (Y \cup W_1) \succ U_2 \cup (X \cup Z)$ or $W_2 \cup (X \cup Z) \succ U_2 \cup (Y \cup W_1)$ **then**
  return CondorcetWinning(refinedPartition)
end if
$(\bar{u}, \bar{w}), (S, S') \leftarrow \text{Uncover}(\mathrm{U}_2, \mathrm{W}_2, (\mathrm{X} \cup \mathrm{Z}), (\mathrm{Y} \cup \mathrm{W}_1))$
Let $\bar{T}$ be indifference class of $\bar{u}$ in $\mathcal{T}$
for $u \in \bar{T} \cap U_1 \cup \{\bar{u}\}$ **do**
  for $w \in W_1 \cup \{\bar{w}\}$ **do**
    $S'' \leftarrow f_{\bar{w},w}(S')$
    if $(S, S'') \notin \mathcal{S}_{u,w}$ **then**
      return CondorcetWinning(refinedPartition)
    end if
    if Compare$((u, w), (S, S''), (X, Y))$ not true **then**
      return CondorcetWinning(refinedPartition)
    end if
    for $z \in Z$ **do**
      for $q \in S'' \cap (W \cup Y)$ **do**
        if Compare$((u, w), (S, S''), (\{z\}, \{q\}))$ not true **then**
          return CondorcetWinning(refinedPartition)
        end if
      end for
    end for
    $(Q, Q') \leftarrow (S \setminus Z \cup \pi_{w^*,w}(W_1), S'' \setminus \pi_{w^*,w}(W_1) \cup Z)$
    if $(Q, Q') \notin S_{u,w}$ **then**

Case 2 $(S, S'') \notin \mathcal{S}_{\bar{u},w}$. Then, either (i) $S \cup \{w\} \succ S'' \cup \{\bar{u}\}$ or (ii) $S'' \cup \{w\} \succ S \cup \{\bar{u}\}$ holds while both is not possible as $\bar{u} \succ w$. First, assume (i) is true. Then, from $(S, S') \in \mathcal{S}_{\bar{u},\bar{w}}$, we know that (iii) $S' \cup \{\bar{u}\} \succ S \cup \{\bar{w}\}$. Reformulating (i) to $S \cup \{w\} \succ S' \setminus \{w\} \cup \{\bar{u}\} \cup \{\bar{w}\}$ and (iii) to $S' \setminus \{w\} \cup \{\bar{u}\} \cup \{w\} \succ S \cup \{\bar{w}\}$ shows that $w \succ \bar{w}$ and in particular $(S, S' \setminus \{w\} \cup \{\bar{u}\}) \in \mathcal{S}_{w,\bar{w}}$. As w and $\bar{w}$ are contained in the same indifference class of $\mathcal{T}$, calling NewCut$(\mathcal{T}, (w, \bar{w}), (S, S' \setminus \{w\} \cup \{\bar{u}\}))$ refines the partition. Second, assume that (ii) holds. However, from $(S, S') \in \mathcal{S}_{\bar{u},\bar{w}}$ we know that (iv) $S \cup \{\bar{u}\} \succ S' \cup \{\bar{w}\}$ is true. As $S'' \cup \{w\} = S' \cup \{\bar{w}\}$ this yields a contradiction to (ii).

We turn to prove that we can find a refined partition within line 24. When Compare$((u, w), (S, S''), (X, Y))$ is not true, then one call to Uncover$(X, Y, S \setminus X, S'' \setminus Y)$ returns a pair (x, y) with $x \succ y$ (or vice versa) and a witness $(P, P') \in \mathcal{S}_{x,y}$ (or $(P, P') \in \mathcal{S}_{y,x}$) (as shown within Lemma 3.14). Since x and y are from the same indifference class of $\mathcal{T}$, namely T_{i_k}, the algorithm can call NewCut$(\mathcal{T}, (x, y), (P, P'))$ and obtain a refined partition.

We continue with the situation in line 29. When Compare$((u, w), (S, S''), (\{z\}, \{w'\}))$ is not true, then a call to Uncover$(\{z\}, \{w'\}, S \setminus \{z\}, S'' \setminus \{w'\})$ returns the pair (z, w') (or (w', z)) and a witness $(P, P') \in \mathcal{S}_{z,w'}$ (or $(P, P') \in \mathcal{S}_{w',z}$). Since z and w' are from the same indifference class of $\mathcal{T}$, namely T_{i_k}, the algorithm can call NewCut$(\mathcal{T}, (z, w'), (P, P'))$ and obtain a refined partition.

We turn to prove that we can find a refined partition within line 35. From $(Q, Q') \notin \mathcal{S}_{u,w}$ we know that either (i) $Q \cup \{w\} \succ Q' \cup \{u\}$ or (ii) $Q' \cup \{w\} \succ Q \cup \{u\}$ while both are not possible as $u \succ w$. First, assume that (i) holds. From $(S, S'') \in \mathcal{S}_{u,w}$ we get in particular that (iii) $S'' \cup \{u\} \succ S \cup \{w\}$ holds. Rewriting (i) as $\pi_{\bar{w},w}(W_1) \cup S \setminus Z \cup \{w\} \succ Z \cup S'' \setminus \pi_{\bar{w},w}(W_1) \cup \{u\}$ and (iii) as $\pi_{\bar{w},w}(W_1) \cup S'' \setminus \pi_{\bar{w},w}(W_1) \cup \{u\} \succ Z \cup S \setminus Z \cup \{w\}$ establishes that we can call Uncover$(\pi_{\bar{w},w}(W_1), Z, S'' \setminus \pi_{\bar{w},w}(W_1) \cup \{u\}, S \setminus Z \cup \{w\})$

which returns a pair $(\hat{w}, \hat{z})$ with $\hat{w} \in \pi_{\bar{w},w}(W_1)$ and $\hat{z} \in Z$ together with a witness for their relation. As $\hat{w}$ and $\hat{z}$ are from the same indifference class of $\mathcal{T}$ we can call NewCut to refine the partition. The case when (ii) follows by an analogous argument.

Lastly, we turn to show that we can find a refined partition within line 40. To see that Compare$((u,w),(Q,Q'),(\{w'\},\{z\}))$ is a valid query, observe that $w' \in Q$ and $z \in Q'$. Moreover, $(Q,Q') \in \mathcal{S}_{u,w}$. Hence, if Compare returns False, then Uncover$(\{w'\},\{z\},Q \setminus \{w'\}, Q' \setminus \{z\})$ returns the pair (w',z) (or (z,w')) together with a witness from $\mathcal{S}_{w',z}$ (or $\mathcal{S}_{z,w'}$). As z and w' are from the same equivalence class of $\mathcal{T}$, we can call the NewCut and obtain a refined partition.

Part II. We now show that the set returned by CondorcetWinning$(\mathcal{T})$ is indeed a Condorcet winning team. If, at some point of the algorithm $i_k \neq i_{2k}$, then the statement follows from Lemma 3.30. Otherwise, the algorithm returns $U \cup X$ which implies that within the last call of CondorcetWinning none of the if conditions was satisfied. We show in the following that this implies that $U \cup X$ is a Condorcet winning team.

We define

$$\begin{aligned} w_1^* &= \operatorname*{argmax}_{w \in W_1 \cup \{\bar{w}\}} v(w), \\ w_2^* &= \operatorname*{argmax}_{w \in W_1 \cup \{\bar{w}\} \setminus \{w_1^*\}} v(w), \text{ and} \\ u^* &= \operatorname*{argmin}_{u \in \bar{T} \cap U_1} v(u). \end{aligned}$$

Moreover, $\epsilon_1 = v(u^*) - v(w_1^*)$ and $\epsilon_2 = v(u^*) - v(w_2^*)$.

We claim that

(i) $|v(X) - v(Y)| < \epsilon_1$, and

(ii) $|v(a) - v(b)| < \epsilon_2$ for all $a \in Y \cup W$ and $b \in Z$.

For (i) observe that there was a point within the iteration of the algorithm when $u = u^*$ and $w = w_1^*$. Moreover, the algorithm called Compare$((u, w), (S, S''), (X, Y))$ which returned true. As we have argued for the subroutine Compare, this implies $\epsilon_1 = v(u^*) - v(w_1^*) > |v(X) - v(Y)|$.

To show (ii), we distinguish three cases. Let $a \in Y \cup W$ and $z \in Z$.

Case 1. $a = w_1^*$. Then, there was a point within the iteration of the algorithm when $u = u^*, w = w_2^*, q = w_1^* = a$ and $z = b$. As Compare$((u, w), (S, S'), (\{z\}, \{q\}))$ returned true in line 28, we know that

$$|v(a) - v(b)| < v(u^*) - v(w_2^*) = \epsilon_2.$$

Case 2. $a \neq w_1^*, a \in S$. Then, there was a point within the iteration of the algorithm when $u = u^*, w = w_1^*, q = a$ and $z = b$. As Compare$((u, w), (S, S'), (\{z\}, \{q\}))$ returned true in line 28, we know that

$$|v(a) - v(b)| < v(u^*) - v(w_1^*) = \epsilon_1 < \epsilon_2.$$

Case 3. $a \neq w_1^*, a \in S'$. Then, there was a point within the iteration of the algorithm when $u = u^*, w = w_1^*, q = a$ and $z = b$. As Compare$((u, w), (Q, Q'), (\{z\}, \{q\}))$ returned true in line 39, we know that

$$|v(a) - v(b)| < v(u^*) - v(w_1^*) = \epsilon_1 < \epsilon_2.$$

Lastly, we show that (i) and (ii) suffice to prove that $U \cup X$ is a Condorcet winning team. To this end let B^* be the best response against $U \cup X$. Observe that $B^* \subseteq Y \cup W \cup Z$.

We start by showing

$$
\begin{aligned}
&v(U \cup X) - v(W \cup Y) \\
&= v(U_1 \cup \{\bar{u}\} \setminus \{u^*\}) + v(u^*) + v(U_2 \setminus \{\bar{u}\}) + v(X) \\
&\quad - v(w_1^*) - v(W_1 \cup \{\bar{w}\} \setminus \{w_1^*\}) - v(W_2 \setminus \{\bar{w}\}) - v(Y) \\
&= v(X) - v(Y) + v(u^*) - v(w_1^*) + v(U_1 \cup \{\bar{u}\} \setminus \{u^*\}) \\
&\quad - v(W_1 \cup \{\bar{w}\} \setminus \{w_1^*\}) + v(U_2 \setminus \{\bar{u}\}) - v(W_2 \setminus \{\bar{w}\}) \\
&> -\epsilon_1 + v(u^*) - v(w_1^*) + v(U_1 \cup \{\bar{u}\} \setminus \{u^*\}) \\
&\quad - v(W_1 \cup \{\bar{w}\} \setminus \{w_1^*\}) + v(U_2 \setminus \{\bar{u}\}) - v(W_2 \setminus \{\bar{w}\}) \\
&> -\epsilon_1 + \epsilon_1 + v(U_1 \cup \{\bar{u}\} \setminus \{u^*\}) - v(W_1 \cup \{\bar{w}\} \setminus \{w_1^*\}) + v(U_2 \setminus \{\bar{u}\}) - v(W_2 \setminus \{\bar{w}\}) \\
&> -\epsilon_1 + \epsilon_1 + |Z| \cdot \epsilon_2 + v(U_2 \setminus \{\bar{u}\}) - v(W_2 \setminus \{\bar{w}\}) \\
&> -\epsilon_1 + \epsilon_1 + |Z| \cdot \epsilon_2 + 0 \\
&= |Z| \cdot \epsilon_2.
\end{aligned}
$$

The first inequality follows by (i), the second by the definition of ϵ_1, the third by the definition of ϵ_2 and the fact that $|u(U_1 \cup \{\bar{u}\} \setminus \{u^*\})| = |v(W_2 \cup \{\bar{w}\} \setminus \{w_1^*\})| = |Z|$, and the last by the fact that $U_2 \triangleright W_2$.

In addition, we get

$$
\begin{aligned}
v(W \cup Y) - v(B^*) &= v(W \cup Y \setminus B^*) - v(B^* \cap Z) \\
&> -|Z| \cdot \epsilon_2,
\end{aligned}
$$

where the inequality follows from the fact that $|v(W \cup Y)| = |v(B^* \cap Z)| < |Z|$ and (ii).

Summing up the two inequalities yields

$$v(U \cup X) - v(B^*) > 0,$$

which concludes this part of the proof.

Part III. It remains to argue about the number of duels performed by *CondorcetWinning1* until it calls *CondorcetWinning2* or returns a team. We first observe that the partition $\mathcal{T}$ can be refined at most $\mathcal{O}(k)$ times. Also, the number of calls to *Uncover* can be bounded by $\mathcal{O}(k)$, since, *Uncover* is either called just before a refinement (hidden within any of the lines saying "refinedPartition") or within line 15. In the following, we will therefore bound the number of duels done within one recursive call of *CondorcetWinning1*. To this end, observe that checking whether some tuple is a subsets witness as well as calling *Compare* requires $\mathcal{O}(1)$ duels. Clearly, the number of times these operations are performed within one recursive call (before the next call is initiated) can be bounded by $\mathcal{O}(k^4)$. Putting all of this together yields that the number of duels can be bounded by $\mathcal{O}(k^5)$. ∎

CondorcetWinning2 We continue by formalizing the second case of the algorithm, which is formalized within Algorithm 8. Since the approach is significantly easier than the one of *CondorcetWinning1*, we directly give the proof.

Lemma 3.30. *After performing* $\mathcal{O}(k^2 \cdot \log(k))$ *many duels,* CondorcetWinning2 *has output a Condorcet winning team.*

Proof. We start by showing that the two duels in line 14 are feasible. To this end observe that U, V, X and Y are disjoint by construction. To argue about their cardinalities, we consider the two cases of the if condition. First, assume $|T_{\leq i_k}| - k <$

Algorithm 9 CordorcetWinning2

Input: a partition of $[n]$ into $T_1 \triangleright T_2 \triangleright \cdots \triangleright T_\ell$ with $i_k \neq i_{2k}$
Output: a CondorcetWinning Team
$j \leftarrow \min\{k - |T_{<i_k}|, |T_{\leq i_k}| - k\}$
Set X and Y to be two disjoint, j-sized subsets of T_{i_k}
Set $W \leftarrow T_{i_k} \setminus X \setminus Y$
Set $L \leftarrow \emptyset$
$(*)$ Set Z to be a subset of $T_{i_{2k}} \setminus L$ of size $2k - |T_{<i_{2k}}|$
while $|L| < |T_{\leq i_{2k}}| - 2k + 1$ **do**
  if $|T_{\leq i_k}| - k < k - |T_{<i_k}|$ **then**
    $U \leftarrow T_{<i_k} \cup W$, $V \leftarrow (T_{>i_k} \cap T_{<i_{2k}}) \cup Z$
  else
    $U \leftarrow T_{<i_k}$, $V \leftarrow W \cup (T_{>i_k} \cap T_{<i_{2k}}) \cup Z$
  end if
  if $V \cup Y \succ U \cup X$ or $V \cup X \succ U \cup Y$ **then**
    return CondorcetWinning(refinedPartition)
  end if
  $(u, v), (S, S') \leftarrow \text{Uncover}(U, V, X, Y)$
  if Compare$((u, v), (S, S'), (X, Y))$ not true **then**
    return CondorcetWinning(refinedPartition)
  end if
  if $v \in Z$ **then**
    $L \leftarrow L \cup \{v\}$, go to $(*)$
  else
    return $U \cup X$
  end if
end while
return $U \cup X$

$k - |T_{<i_k}|$. Then

$$|U| = |T_{<i_k}| + |T_{i_k}| - 2j = |T_{\leq i_k}| - (|T_{\leq i_k}| - k) - j = k - j.$$

As $|X| = |Y| = j$, we get that $|U| + |X| = |U| + |Y| = k$. Similarly, for the other case, we have

$$|V| = |T_{<i_{2k}}| - |T_{\leq i_k}| + |Z| = |T_{<i_{2k}}| - |T_{\leq i_k}| + 2k = |T_{<i_k}| = k + k - |T_{i_k}| = k - j.$$

Hence, also $|U| + |X| = |U| + |Y| = k$.

Next, we show that we can find a refined partition in line 15. Assume wlog that $V \cup Y \succ U \cup X$ holds and observe that both statements cannot be true as $U \triangleright V$ by construction. Hence, we have $U \cup Y \succ V \cup X$ which implies that we can call Uncover(Y, X, U, V) which returns a pair (y, x) as well as a witness from $\mathcal{S}_{y,x}$ (or $\mathcal{S}_{x,y}$). Since x and y are from the same indifference class of $\mathcal{T}$, namely T_{i_k}, we can call the NewCut subroutine and obtain a refined partition.

The call to Uncover in line 17 is feasible, as the non-satisfaction of the if condition implies that $U \cup X \succ V \cup Y$ and $U \cup Y \succ V \cup X$.

In line 19 we can refine the partition $\mathcal{T}$, as, if Compare$((u, v), (S, S'), (X, Y))$ does not return true, then Uncover$(X, Y, S \setminus X, S' \setminus Y)$ returns a pair (x, y) with $x \in X$ and $y \in Y$ (or (y, x)) together with a witness from $\mathcal{S}_{x,y}$ (or $\mathcal{S}_{y,x}$). Since x and y are both from the same indifference class of $\mathcal{T}$, namely T_{i_k}, we can refine $\mathcal{T}$ by calling the NewCut subroutine.

Lastly, we show that $U \cup X$ is a Condorcet winning team when the algorithm reaches line 24 or line 27. We first discuss line 24. First, observe that $U \triangleright V, u \in U$, $v \in V$ and (S, S') is a witness for their relation, that is, $(S, S') \in \mathcal{S}_{uv}$. Moreover, since Compare$((u, v), (S, S'), (X, Y))$ is true, we know that

$$v(u) - v(v) > |v(X) - v(Y)|. \tag{3.12}$$

Additionally we know that $v \in V \setminus Z$, which implies that $v \in T_{<i_{2k}}$. Hence, v is in particular contained in the best response against $U \cup X$. Since Y is also guaranteed to be within the best response, we can denote the best response by $V' \cup Y$. Using

(3.12) and the fact that $U \triangleright V'$, we get

$$\begin{aligned} v(U \cup X) - v(V' \cup Y) &= v(U \setminus \{u\}) + v(u) + v(X) - v(V' \setminus \{v\}) - v(v) - v(Y) \\ &> v(U \setminus \{u\}) - v(V' \setminus \{v\}) > 0, \end{aligned}$$

showing that $U \cup X \succ V' \cup Y$.

Now, consider the situation in line 27. This implies that the list L is of length $|T_{\leq i_{2k}}| - 2k + 1$ and for each $v \in L$ there exists $u \in U$ such that

$$v(u) - v(v) > |v(X) - v(Y)|. \tag{3.13}$$

Again, the best response against $U \cup X$ contains Y. Denote the best response by $V' \cup Y$. By the size of L we know that $V' \cap L \neq \emptyset$. Let v be a node in the intersection and u be the node for which the algorithm has proven (3.13). Due to the same argumentation as before, $U \triangleright V'$ and $v(u') - v(v') > v(X) - v(Y)$ implies $U \cup X \succ V' \cup Y$.

It remains to argue about the number of duels performed by *CondorcetWinning2*. Again, it is clear that the partition $\mathcal{T}$ can be refined at most $\mathcal{O}(k)$ times. Per refinement, the is one additional call to *Uncover* which is bounded by $\mathcal{O}(\log(k))$ duels. Moreover, the iterations of the while loop can be bounded by $\mathcal{O}(k)$. Within one iteration the algorithm performs *Compare* (requiring $\mathcal{O}(1)$ duels) and *Uncover* (requiring $\mathcal{O}(\log(k))$ duels). Putting everything together the number of duels can hence be bounded by $\mathcal{O}(k^2 \log(k))$. ∎

Putting Lemma 3.29 and Lemma 3.30 together clearly yields the proof of Lemma 3.15.

Lemma 3.15. *After performing $\mathcal{O}(k^5)$ many duels,* CondorcetWinning1 *has identified a Condorcet winning team.* CondorcetWinning2 *identifies a Condorcet winning team*

after $\mathcal{O}(k^2 \log(k))$ *duels.*

Extension to a Stochastic Environment In the following we sketch how we can reduce any stochastic instance satisfying $|P_{A,B} - 1/2| \in [1/2+\theta, 1]$ to our deterministic setting. To achieve such a reduction, simulate each deterministic duel by $\mathcal{O}(\frac{\ln m/\delta}{\theta^2})$ stochastic duels to determine the duel's winner with probability at least $1 - \delta/m$, where $\mathcal{O}(m)$ is the sample complexity of an algorithm that finds a Condorcet winning team in the deterministic case. An invocation of Chernoff-Hoeffding concentration bound yields that each duel's winner is correctly determined by this simulation with probability at least $1-\delta/m$, and applying union bound over the total number of duels results in an algorithm that requires $\mathcal{O}(m\frac{\ln m/\delta}{\theta^2})$ team duels to identify a Condorcet winning team with probability at least $1 - \delta$.

Additive Total Orders

In the following we provide a sufficient condition for assigning values to players in a way that complies with a total order on teams, assuming that each team has value of the cumulative values of it's players and that team A is better than team B if and only if the value of A is larger than the value of B. Formally:

Given: A set of players $[n]$ and a total order $\succ$ on the subsets of size k.

Question: Do there exist values for the players representing this order? Or more precisely, does the following system of linear inequalities have a feasible solution?

We denote define $\mathcal{D} = \{(A, B) \mid A, B \text{ are teams, } A \succ B\}$.

$$\sum_{b \in B} x_b - \sum_{a \in A} x_a \leq -1 \text{ for all } (A, B) \in \mathcal{D}$$

$$x_a \geq 0 \text{ for all } a \in [n]$$

We remark that, alternatively to -1 on the right hand side, we could have chosen any other negative number.

The following is a variant of Farkas Lemma:

Lemma 3.31 (Farkas' Lemma [54])**.** *Let $n, m \in \mathbb{N}$, $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^m$. Then, exactly one of the following is true.*

1. $\exists\, x \in \mathbb{R}^n, Ax \leq b, x \geq 0$
2. $\exists\, y \in \mathbb{R}^m, y^T A \geq 0, y \geq 0$ *and* $y^T b < 0$.

Imagine the system above in matrix form Ax, then the system $y^T A \geq 0, y^T b < 0, y \geq 0$ looks as follows:

$$\sum_{(A,B)\in\mathcal{D}:i\in B} y_{AB} - \sum_{(A,B)\in\mathcal{D}:i\in A} y_{AB} \geq 0 \text{ for all players } i \in [n]$$

$$y_{AB} \geq 0 \text{ for all } (A,B) \in \mathcal{D}$$

$$\sum_{(A,B)\in\mathcal{D}} y_{AB} > 0$$

Assume the second system does have a feasible solution $y \geq 0$. In particular, there exists one pair $A \succ B$ for which $y_{AB} > 0$. We can assume wlog that this solution is rational and by scaling it up that it is integer.

We define the following condition:

Condition (*) There exist $\mathcal{A} = \{A_1, \ldots, A_m\}$ and $\mathcal{B} = \{B_1, \ldots, B_m\}$ satisfying the following two conditions:

(i) $A_j \succ B_j$ for all $j \in [m]$

(ii) Let $n_i^{\mathcal{A}}$ be the number of times that player i is included in some element of $\mathcal{A}$. Define $n_i^{\mathcal{B}}$ analogously. Then, $n_i^{\mathcal{A}} = n_i^{\mathcal{B}}$ for all players $i \in [n]$.

Claim 3.32. *The second system of linear inequalities has a feasible solution if and only if* $(*)$ *is satisfied.*

Proof. " $\Rightarrow$ " Assume the second system has a feasible (and wlog integral) solution y. We construct $\mathcal{A}$ and $\mathcal{B}$ as follows: For each pair $A \succ B$ for which $y_{AB} > 0$, add exactly y_{AB} copies of A and B to $\mathcal{A}$ and $\mathcal{B}$, respectively. The first constraints for condition $(*)$ is clearly satisfied. Now, assume for contradiction that there exists a player $i \in [n]$ for which $n_i^{\mathcal{A}} > n_i^{\mathcal{B}}$ holds. Then, we get

$$\sum_{(A,B)\in\mathcal{D}:i\in B} y_{AB} - \sum_{(A,B)\in\mathcal{D}:i\in A} y_{AB} = n_i^{\mathcal{B}} - n_i^{\mathcal{A}} < 0,$$

a contradiction to the feasibility of y. On the other hand, assume that there exists a player $i \in [n]$ for which $n_i^{\mathcal{A}} < n_i^{\mathcal{B}}$ holds. Observe that

$$\sum_{j\in[n]} n_j^{\mathcal{A}} = \sum_{j\in[n]} n_j^{\mathcal{B}} = |\mathcal{A}|k$$

and hence

$$\sum_{j\in[n]\setminus\{i\}} n_j^{\mathcal{A}} > \sum_{j\in[n]\setminus\{i\}} n_j^{\mathcal{B}},$$

which implies that there exists some $i' \in [n] \setminus \{i\}$ with $n_{i'}^{\mathcal{A}} > n_{i'}^{\mathcal{B}}$, a contradiction.

" $\Leftarrow$ " Assume that there exist $\mathcal{A}$ and $\mathcal{B}$ satisfying condition $(*)$. Then, set $y_{A_j,B_j} = |\{q \in [m] : (A_q, B_q) = (A_j, B_j)\}|$ for all $j \in [m]$ and $y_{A,B} = 0$ for all other duels. This is a feasible solution to the second system of inequalities. ∎

This directly yields the sufficient condition for a total order to be representable by

values.

Corollary 3.33. *There exists a solution to the first system of inequalities if and only if condition* $(*)$ *does not hold.*

3.2 Departing Bandits

3.2.1 Introduction

At the heart of online services spanning such diverse industries as media consumption, dating, financial products, and more, recommendation systems (RSs) drive personalized experiences by making curation decisions informed by each user's past history of interactions. While in practice, these systems employ diverse statistical heuristics, much of our theoretical understanding of them comes via stylized formulations within the multi-armed bandits (MABs) framework. While MABs abstract away from many aspects of real-world systems they allow us to extract crisp insights by formalizing fundamental tradeoffs, such as that between exploration and exploitation that all RSs must face [73, 94, 103, 114]. As applies to RSs, exploitation consists of continuing to recommend items (or categories of items) that have been observed to yield high rewards in the past, while exploration consists of recommending items (or categories of items) about which the RS is uncertain but that could *potentially* yield even higher rewards.

In traditional formalizations of RSs as MABs, the recommender's decisions affect only the rewards obtained. However, real-life recommenders face a dynamic that potentially alters the exploration-exploitation tradeoff: Dissatisfied users have the option to depart the system, never to return. Thus, recommendations in the service of exploration not only impact instantaneous rewards but also risk driving away users and therefore can influence long-term cumulative rewards by shortening trajectories

of interactions.

In this work, we propose *departing bandits* which augment conventional MABs by incorporating these policy-dependent horizons. To motivate our setup, we consider the following example: An RS for recommending blog articles must choose at each time among two categories of articles, e.g., economics and sports. Upon a user's arrival, the RS recommends articles sequentially. After each recommendation, the user decides whether to "click" the article and continue to the next recommendation, or to "not click" and may leave the system. Crucially, the user interacts with the system for a random number of rounds. The user's departure probability depends on their satisfaction from the recommended item, which in turn depends on the user's unknown *type*. A user's type encodes their preferences (hence the probability of clicking) on the two topics (economics and sports).

When model parameters are given, in contrast to traditional MABs where the optimal policy is to play the best fixed arm, departing bandits require more careful analysis to derive an optimal planning strategy. Such planning is a local problem, in the sense that it is solved for each user. Since the user type is never known explicitly (the recommender must update its beliefs over the user types after each interaction), finding an optimal recommendation policy requires solving a specific partially observable MDP (POMDP) where the user type constitutes the (unobserved) state (more details in Section 3.2.5). When the model parameters are unknown, we deal with a learning problem that is global, in the sense that the recommender (learner) is learning for a stream of users instead of a particular user.

We begin with a formal definition of departing bandits in Section 3.2.2, and demonstrate that any fixed-arm policy is prone to suffer linear regret. In Section 3.2.3, we establish the UCB-based learning framework used in later sections. We instantiate this framework with a single user type in Section 3.2.4, where we show that it achieves

$\tilde{O}(\sqrt{T})$ regret for T being the number of users. We then move to the more challenging case with two user types and two recommendation categories in Section 3.2.5. To analyze the planning problem, we effectively reduce the search space for the optimal policy by using a closed-form of the *expected return* of any recommender policy. These results suggest an algorithm that achieves $\tilde{O}(\sqrt{T})$ regret in this setting. Finally, we also show an efficient optimal planning algorithm for multiple user types and two recommendation categories, and describe a scheme to construct semi-synthetic problem instances for this setting using real-world datasets.

Related Work

MABs have been studied extensively by the online learning community [31, 25]. The contextual bandit literature augments the MAB setup with context-dependent rewards [1, 121, 97, 86, 88]. In contextual bandits, the learner observes a *context* before they make a decision, and the reward depends on the context. Another line of related work considers the dynamics that emerge when users act strategically [87, 98, 40, 10, 9]. In that line of work, users arriving at the system receive a recommendation but act strategically: They can follow the recommendation or choose a different action. This modeling motivates the development of incentive-compatible mechanisms as solutions. In our work, however, the users are modeled in a stochastic (but not strategic) manner. Users may leave the system if they are dissatisfied with recommendations, and this departure follows a fixed (but possibly unknown) stochastic model.

The departing bandits problem has two important features: Policy-dependent horizons, and multiple user types that can be interpreted as unknown states. Existing MAB works [8, 29] have addresses these phenomena separately but we know of no work that integrates the two in a single framework. In particular, while [8] study the setting with multiple user types, they focus on a fixed horizon setting. Additionally,

while [29] deal with departure probabilities and policy-dependent interaction times for a single user type, they do not consider the possibility of multiple underlying user types.

The planning part of our problem falls under the framework of using Markov Decision Processes for modeling recommender-user dynamics [119]. Specifically, our problem works with partially observable user states which have also been seen in many recent bandits variants [108, 90]. Unlike these prior works that focus on interactions with a single user, departing bandits consider a stream of users each of which has an (unknown) type selected among a finite set of user types.

More broadly, our RS learning problem falls under the domain of reinforcement learning (RL). Existing RL literature that considers departing users in RSs include [136, 96, 135]. While [136] handle users of a single type that depart the RS within a bounded number of interactions, our work deals with multiple user types. In contrast to [135], we consider an online setting and provide regret guarantees that do not require bounded horizon. Finally, [96] use POMDPs to model user departure and focus on approximating the value function. They conduct an experimental analysis on historical data, while we devise an online learning algorithm with theoretical guarantees.

3.2.2 Departing Bandits: Problem Formulation

We propose a new online problem, called *departing bandits*, where the goal is to find the optimal recommendation algorithm for users of (unknown) types, and where the length of the interactions depends on the algorithm itself. Formally, the departing bandits problem is defined by a tuple $\langle [M], [K], \mathbf{q}, \mathbf{P}, \mathbf{\Lambda} \rangle$, where M is the number of user *types*, K is the number of *categories*, $\mathbf{q} \in [0,1]^M$ specifies a prior distribution over types, and $\mathbf{P} \in (0,1)^{K \times M}$ and $\mathbf{\Lambda} \in (0,1)^{K \times M}$ are the *click-probability* and the

departure-probability matrices, respectively.[6]

There are T users who arrive sequentially at the RS. At every episode, a new user $t \in [T]$ arrives with a type $type(t)$. We let $\mathbf{q}$ denote the prior distribution over the user types, i.e., $type(t) \sim \mathbf{q}$. Each user of type x *clicks* on a recommended category a with probability $\mathbf{P}_{a,x}$. In other words, each click follows a Bernoulli distribution with parameter $\mathbf{P}_{a,x}$. Whenever the user clicks, she stays for another iteration, and when the user does not click (no-click), she *departs* with probability $\mathbf{\Lambda}_{a,x}$ (and stays with probability $1 - \mathbf{\Lambda}_{a,x}$). Each user t interacts with the RS (the learner) until she departs.

We proceed to describe the user-RS interaction protocol. In every iteration j of user t, the learner recommends a category $a \in [K]$ to user t. The user clicks on it with probability $\mathbf{P}_{a,type(t)}$. If the user clicks, the learner receives a reward of $r_{t,j}(a) = 1$.[7] If the user does not click, the learner receives no reward (i.e., $r_{t,j}(a) = 0$), and user t departs with probability $\mathbf{\Lambda}_{a,type(t)}$. We assume that the learner knows the value of a constant $\epsilon > 0$ such that $\max_{a,x} \mathbf{P}_{a,x} \leq 1 - \epsilon$ (i.e., ϵ does not depend on T). When user t departs, she does not interact with the learner anymore (and the learner moves on to the next user $t + 1$). For convenience, the departing bandits problem protocol is summarized in Algorithm 10.

Having described the protocol, we move on to the goals and performance of the learner. Without loss of generality, we assume that the online learner's recommendations are made based on a *policy* π, which is a mapping from the history of previous interactions (with that user) to recommendation categories. For each user (episode) $t \in [T]$, the learner selects a policy π_t that recommends category $\pi_{t,j} \in [K]$ at every iteration

[6] We denote by $[n]$ the set $\{1, \ldots, n\}$.

[7] We formalize the reward as is standard in the online learning literature, from the perspective of the learner. However, defining the reward from the user perspective by, e.g., considering her utility as the number of clicks she gives or the number of articles she reads induces the same model.

Algorithm 10 The Departing Bandits Protocol

Input: number of types M, number of categories K, and number of users (episodes) T
Hidden Parameters: types prior $\mathbf{q}$, click-probability $\mathbf{P}$, and departure-probability $\mathbf{\Lambda}$

for episode $t \leftarrow 1, \ldots, T$ **do**
  a new user with type $type(t) \sim \mathbf{q}$ arrives
  $j \leftarrow 1$, $depart \leftarrow false$
  while *depart* is $false$ **do**
    the learner picks a category $a \in [K]$
    with probability $\mathbf{P}_{a,x}$, user t clicks on a and $r_{t,j}(a) \leftarrow 1$; otherwise, $r_{t,j}(a) \leftarrow 0$
    if $r_{t,j}(a) = 0$ **then**
      with probability $\mathbf{\Lambda}_{a,x}$: $depart \leftarrow true$ and user t departs
    end if
    the learner observes $r_{t,j}(a)$ and *depart*
    if *depart* is $false$ **then**
      $j \leftarrow j + 1$
    end if
  end while
end for

$j \in [N^{\pi_t}(t)]$, where $N^{\pi_t}(t)$ denotes the episode length (i.e., total number of iterations policy π_t interacts with user t until she departs).[8] The *return* of a policy π, denoted by V^π is the cumulative reward the learner obtains when executing the policy π until the user departs. Put differently, the return of π from user t is the random variable $V^\pi = \sum_{j=1}^{N^\pi(t)} r_{t,j}(\pi_{t,j})$.

We denote by π^* an optimal policy, namely a policy that maximizes the expected return, $\pi^* = \text{argmax}_\pi \, \mathbb{E}[V^\pi]$. Similarly, we denote by V^* the optimal return, i.e., $V^* = V^{\pi^*}$.

We highlight two algorithmic tasks. The first is the planning task, in which the goal is to find an optimal policy π^*, given $\mathbf{P}, \mathbf{\Lambda}, \mathbf{q}$. The second is the online learning task. We consider settings where the learner knows the number of categories, K,

[8] We limit the discussion to deterministic policies solely; this is w.l.o.g. (see Subsection 3.2.5 for further details).

	Type x	Type y
Category 1	$\mathbf{P}_{1,x} = 0.5$	$\mathbf{P}_{1,y} = 0.28$
Category 2	$\mathbf{P}_{2,x} = 0.4$	$\mathbf{P}_{2,y} = 0.39$
Prior	$\mathbf{q}_x = 0.4$	$\mathbf{q}_y = 0.6$

Table 3.1: The departing bandits instance in Section 3.2.2.

the number of types, M, and the number of users, T, but has no prior knowledge regarding $\mathbf{P}, \mathbf{\Lambda}$ or $\mathbf{q}$. In the online learning task, the *value* of the learner's algorithm is the sum of the returns obtained from all the users, namely

$$\sum_{t=1}^{T} V^{\pi_t} = \sum_{t=1}^{T} \sum_{j=1}^{N^{\pi}(t)} r_{t,j}(\pi_{t,j}).$$

The performance of the leaner is compared to that of the best policy, formally defined by the *regret* for T episodes,

$$R_T = T \cdot \mathbb{E}[V^{\pi^*}] - \sum_{t=1}^{T} V^{\pi_t}. \tag{3.14}$$

The learner's goal is to minimize the expected regret $\mathbb{E}[R_T]$.

Example

The motivation for the following example is two-fold. First, to get the reader acquainted with our notations; and second, to show why fixed-arm policies are inferior in our setting.

Consider a problem instance with two user types ($M = 2$), which we call x and y for convenience. There are two categories ($K = 2$), and given no-click the departure is deterministic, i.e., $\mathbf{\Lambda}_{a,\tau} = 1$ for every category $a \in [K]$ and type $\tau \in [M]$. That is, every user leaves immediately if she does not click. Furthermore, let the click-probability $\mathbf{P}$ matrix and the user type prior distribution $\mathbf{q}$ be as in Table 3.1.

Looking at $\mathbf{P}$ and $\mathbf{q}$, we see that Category 1 is better for Type x, while Category 2 is better for type y. Notice that without any additional information, a user is more likely to be type y. Given the prior distribution, recommending Category 1 in the first round yields an expected reward of $\mathbf{q}_x\mathbf{P}_{1,x} + \mathbf{q}_y\mathbf{P}_{1,y} = 0.368$. Similarly, recommending Category 2 in the first round results in an expected reward of 0.394. Consequently, if we recommend *myopically*, i.e., without considering the user type, always recommending Category 2 is better than always recommending Category 1.

Let π^a denote the fixed-arm policy that always selects a single category a. Using the tools we derive in Section 3.2.5 and in particular Theorem 3.42, we can compute the expected returns of π^1 and π^2, $\mathbb{E}[V^{\pi^1}]$ and $\mathbb{E}[V^{\pi^2}]$. Additionally, using results from Section 3.2.5, we can show that the optimal policy for the planning task, π^*, recommends Category 2 until iteration 7, and then recommends Category 1 for the rest of the iterations until the user departs.

Using simple calculations, we see that $\mathbb{E}[V^{\pi^*}] - \mathbb{E}[V^{\pi^1}] > 0.0169$ and $\mathbb{E}[V^{\pi^*}] - \mathbb{E}[V^{\pi^2}] > 1.22 \times 10^{-5}$; hence, the expected return of the optimal policy is greater than the returns of both fixed-arm policies by a constant. As a result, if the learner only uses fixed-arm policies (π^a for every $a \in [K]$), she suffers linear expected regret, i.e., $\mathbb{E}[R_T] = T \cdot \mathbb{E}[V^{\pi^*}] - \sum_{t=1}^{T} \mathbb{E}[V^{\pi^a}] = \Omega(T)$.

3.2.3 UCB Policy for Sub-exponential Returns

In this section, we introduce the learning framework used in this section and provide a general regret guarantee for it.

In standard MAB problems, at each $t \in [T]$ the learner picks a single arm and receives a single sub-Gaussian reward. In contrast, in departing bandits, at each $t \in [T]$ the

learner receives a return V^π, which is the cumulative reward of that policy. The return V^π depends on the policy π not only through the obtained rewards at each iteration but also through the total number of iterations (trajectory length). Such returns are not necessarily sub-Gaussian. Consequently, we cannot use standard MAB algorithms as they usually rely on concentration bounds for sub-Gaussian rewards. Furthermore, as we have shown in Section 3.2.2, in departing bandits fixed-arm policies can suffer linear regret (in terms of the number of users), which suggests considering a more expressive set of policies. This in turn yields another disadvantage for using MAB algorithms for departing bandits, as their regret is linear in the number of arms (categories) K.

As we show later in Sections 3.2.4 and 3.2.5, for some natural instances of the departing bandits problem, the return from each user is sub-exponential (Definition 3.34). Algorithm 11, which we propose below, receives a set of policies Π as input, along with other parameters that we describe shortly. The algorithm is a restatement of the *UCB-Hybrid* Algorithm from [72], with two modifications: (1) The input includes a set of policies rather than a set of actions/categories, and accordingly, the confidence bound updates are based on return samples (denoted by $\hat{V}^\pi$) rather than reward samples. (2) There are two global parameters ($\tilde{\tau}$ and η) instead of two local parameters per action. If the return from each policy in Π is sub-exponential, Algorithm 11 not only handles sub-exponential returns, but also comes with the following guarantee: Its expected value is close to the value of the best policy in Π.

Sub-exponential Returns

For convenience, we state here the definition of sub-exponential random variables [48].

Definition 3.34. *We say that a random variable X is* sub-exponential *with parameters (τ^2, b) if for every γ such that $|\gamma| < 1/b$,*

$$\mathbb{E}[\exp(\gamma(X - \mathbb{E}[X]))] \leq \exp(\frac{\gamma^2\tau^2}{2}).$$

In addition, for every (τ^2, b)-sub-exponential random variables, there exist constants $C_1, C_2 > 0$ such that the above is equivalent to each of the following properties:

1. *Tails:* $\forall v \geq 0 : \Pr[|X| > v] \leq \exp(1 - \frac{v}{C_1})$.
2. *Moments:* $\forall p \geq 1 : (\mathbb{E}[|X|^p])^{1/p} \leq C_2 p$.

Let Π be a set of policies with the following property: There exist $\tilde{\tau}, \eta$ such that the return of every policy $\pi \in \Pi$ is (τ^2, b)-sub-exponential with $\tilde{\tau} \geq \tau$ and $\eta \geq \frac{b^2}{\tau^2}$. The following Algorithm 11 receives as input a set of policies Π with the associated parameters, $\tilde{\tau}$ and η. Similarly to the UCB algorithm, it maintains an upper confidence bound U for each policy, and balances between exploration and exploitation. Theorem 3.35 below shows that Algorithm 11 always gets a value similar to that of the best policy in Π up to an additive factor of $\tilde{O}\left(\sqrt{|\Pi|T} + |\Pi|\right)$. The theorem follows directly from Theorem 3 from [72] by having policies as arms and returns as rewards.

Theorem 3.35. *Let Π be a set of policies with the associated parameters $\tilde{\tau}, \eta$. Let $\pi_1, \ldots, \pi_T$ be the policies Algorithm 11 selects. It holds that*

$$\mathbb{E}\left[\max_{\pi \in \Pi} T \cdot V^{\pi} - \sum_{t=1}^{T} V^{\pi_t}\right] = O(\sqrt{|\Pi| T \log T} + |\Pi| \log T).$$

There are two challenges in leveraging Theorem 3.35. The first challenge is crucial: Notice that Theorem 3.35 does not imply that Algorithm 11 has a low regret; its only guarantee is w.r.t. the policies in Π received as an input. As the number of policies is infinite, our success will depend on our ability to characterize a "good" set

Algorithm 11 UCB-based algorithm with hybrid radii: UCB-Hybrid [72]

Input: set of policies Π, number of users T, $\tilde{\tau}, \eta$
Initialize: $\forall \pi \in \Pi : U_0(\pi) \leftarrow \infty, n(\pi) = 0$
for user $t \leftarrow 1, \ldots, T$ **do**
  Execute π_t such that $\pi_t \in \text{argmax}_{\pi \in \Pi} U_{t-1}(\pi)$ and receive return $\hat{V}^{\pi_t}[n(\pi_t)] \leftarrow \sum_{j=1}^{N^{\pi_t}(t)} r_{t,j}(\pi_{t,j})$
  $n(\pi_t) \leftarrow n(\pi_t) + 1$
  if $n(\pi_t) < 8\eta \ln T$ **then**
    Update $U_t(\pi_t) = \frac{\sum_{i=1}^{n(\pi_t)} \hat{V}^{\pi_t}[i]}{n(\pi_t)} + \frac{8\sqrt{\eta} \cdot \tilde{\tau} \ln T}{n(\pi_t)}$
  else
    Update $U_t(\pi_t) = \frac{\sum_{i=1}^{n(\pi_t)} \hat{V}^{\pi_t}[i]}{n(\pi_t)} + \sqrt{\frac{8\tilde{\tau}^2 \ln T}{n(\pi_t)}}$
  end if
end for

of policies Π. The second challenge is technical: Even if we find such Π, we still need to characterize the associated $\tilde{\tau}$ and η. This is precisely what we do in Section 3.2.4 and 3.2.5.

3.2.4 Single User Type

In this section, we focus on the special case of a single user type, i.e., $M = 1$. For notational convenience, since we only discuss single-type users, we associate each category $a \in [K]$ with its two unique parameters $\mathbf{P}_a := \mathbf{P}_{a,1}, \mathbf{\Lambda}_a := \mathbf{\Lambda}_{a,1}$ and refer to them as scalars rather than vectors. In addition, We use the notation N_a for the random variable representing the number of iterations until a random user departs after being recommended by π^a, the fixed-arm policy that recommends category a in each iteration.

To derive a regret bound for single-type users, we use two main lemmas: Lemma 3.36, which shows the optimal policy is fixed, and Lemma 3.38, which shows that returns of fixed-arm policies are sub-exponential and calculate their corresponding parameters. These lemmas allow us to use Algorithm 11 with a policy set Π that contains all the

fixed-arm policies, and derive a $\tilde{O}(\sqrt{T})$ regret bound.

To show that there exists a category $a^* \in [K]$ for which π^{a^*} is optimal, we rely on the assumption that all the users have the same type (hence we drop the type subscripts t), and as a result the rewards of each category $a \in [K]$ have an expectation that depends on a single parameter, namely $\mathbb{E}[r(a)] = \mathbf{P}_a$. Such a category $a^* \in [K]$ does not necessarily have the maximal click-probability nor the minimal departure-probability, but rather an optimal combination of the two (in a way, this is similar to the knapsack problem, where we want to maximize the reward while having as little weight as possible). We formalize it in the following lemma.

Lemma 3.36. *A policy π^{a^*} is optimal if*

$$a^* \in \operatorname*{argmax}_{a \in [K]} \frac{\mathbf{P}_a}{\mathbf{\Lambda}_a(1 - \mathbf{P}_a)}.$$

As a consequence of this lemma, the planning problem for single-type users is trivial—the solution is a fixed-arm policy π^{a^*} given in the lemma. However, without access to the model parameters, identifying π^{a^*} requires learning. We proceed with a simple observation regarding the random number of iterations obtained by executing a fixed-arm policy. The observation would later help us show that the return of any fixed-arm policy is sub-exponential.

Observation 3.37. *For every $a \in [K]$ and every $\mathbf{\Lambda}_a > 0$, the random variable N_a follows a geometric distribution with success probability parameter $\mathbf{\Lambda}_a[1 - \mathbf{P}_a] \in (0, 1 - \epsilon]$.*

Using Observation 3.37 and previously known results (stated as Lemma 3.53 in Section 3.2.4), we show that N_a is sub-exponential for all $a \in [K]$. Notice that return realizations are always upper bounded by the trajectory length; this implies that

returns are also sub-exponential. However, to use the regret bound of Algorithm 11, we need information regarding the parameters (τ_a^2, b_a) for every policy π^a. We provide this information in the following Lemma 3.38.

Lemma 3.38. *For each category $a \in [K]$, the centred random variable $V^{\pi^a} - \mathbb{E}[V^{\pi^a}]$ is sub-exponential with parameters (τ_a^2, b_a), such that*

$$\tau_a = b_a = -\frac{8e}{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))}.$$

Proof sketch. We rely on the equivalence between the subexponentiality of a random variable and the bounds on its moments (Property 2 in Definition 3.34). We bound the expectation of the return V^{π^a}, and use Minkowski's and Jensen's inequalities to show in Lemma 3.52 that $\mathbb{E}[|V^{\pi^a} - \mathbb{E}[V^{\pi^a}]|^p])^{1/p}$ is upper bounded by $-4/\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))$ for every $a \in [K]$ and $p \geq 1$. Finally, we apply a normalization trick and bound the Taylor series of $\mathbb{E}[\exp(\gamma(V^{\pi^a} - \mathbb{E}[V^{\pi^a}]))]$ to obtain the result. ∎

An immediate consequence of Lemma 3.38 is that the parameters $\tilde{\tau} = 8e/\ln(\frac{1}{1-\epsilon})$ and $\eta = 1$ are valid upper bounds for τ_a and b_a/τ_a^2 for each $a \in [K]$ (I.e., $\forall a \in [K]$: $\tilde{\tau} \geq \tau_a$ and $\eta \geq b_a^2/\tau_a^2$). We can now derive a regret bound using Algorithm 11 and Theorem 3.35.

Theorem 3.39. *For single-type users ($M = 1$), running Algorithm 11 with $\Pi = \{\pi^a : a \in [K]\}$ and $\tilde{\tau} = \frac{8e}{\ln(\frac{1}{1-\epsilon})}$, $\eta = 1$ achieves an expected regret of at most*

$$\mathbb{E}[R_T] = O(\sqrt{KT \log T} + K \log T).$$

3.2.5 Two User Types and Two Categories

In this section, we consider cases with two user types ($M = 2$), two categories ($K = 2$) and departure-probability $\mathbf{\Lambda}_{a,\tau} = 1$ for every category $a \in [K]$ and type $\tau \in [M]$.

Even in this relatively simplified setting, where users leave after the first "no-click", planning is essential. To see this, notice that the event of a user clicking on a certain category provides additional information about the user, which can be used to tailor better recommendations; hence, algorithms that do not take this into account may suffer a linear regret. In fact, this is not just a matter of the learning algorithm at hand, but rather a failure of all fixed-arm policies; there are instances where all fixed-arm policies yield high regret w.r.t. the baseline defined in Equation (3.14). Indeed, this is what the example in Section 3.2.2 showcases. Such an observation suggests that studying the optimal planning problem is vital.

In Section 3.2.5, we introduce the partially observable MDP formulation of departing bandits along with notion of *belief-category walk*. We use this notion to provide a closed-form formula for policies' expected return, which we use extensively later on. Next, in Section 3.2.5 we characterize the optimal policy, and show that we can compute it in constant time relying on the closed-form formula. This is striking, as generally computing optimal POMDP policies is computationally intractable since, e.g., the space of policies grows exponentially with the horizon. Conceptually, we show that there exists an optimal policy that depends on a belief threshold: It recommends one category until the posterior belief of one type, which is monotonically increasing, crosses the threshold, and then it recommends the other category. Finally, in Section 3.2.5 we leverage all the previously obtained results to derive a small set of threshold policies of size $O(\ln T)$ with corresponding sub-exponential parameters. Due to Theorem 3.35, this result implies a $\tilde{O}(\sqrt{T})$ regret.

Efficient Planning

To recap, we aim to find the optimal policy when the click-probability matrix and the prior over user types are known. Namely, given an instance in the form of $\langle \mathbf{P}, \mathbf{q} \rangle$,

our goal is to efficiently find the optimal policy.

For planning purposes, the problem can be modeled by an episodic POMDP, $\langle S, [K], O, \mathrm{Tr}, \mathbf{P}, \Omega, \mathbf{q}, O \rangle$. A set of states, $S = [M] \cup \{\perp\}$ that comprises all types $[M]$, along with a designated absorbing state $\perp$ suggesting that the user departed (and the episode terminated). $[K]$ is the set of the actions (categories). $O = \{stay, depart\}$ is the set of possible observations. The transition and observation functions, $\mathrm{Tr} : S \times [K] \to S$ and $\Omega : S \times [K] \to O$ (respectively) satisfy $\mathrm{Tr}(\perp | i, a) = \Omega(depart | i, a) = 1 - \mathbf{P}_{i,a}$ and $\mathrm{Tr}(i|i, a) = \Omega(stay|i, a) = \mathbf{P}_{i,a}$ for every type $i \in [M]$ and action $a \in [K]$. Finally, $\mathbf{P}$ is the expected reward matrix, and $\mathbf{q}$ is the initial state distribution over the M types.

When there are two user types and two categories, the click-probability matrix is given by Table 3.2 where we note that the prior on the types holds $\mathbf{q}_y = 1 - \mathbf{q}_x$, thus can be represented by a single parameter $\mathbf{q}_x$.

Remark 3.40. *Without loss of generality, we assume that* $\mathbf{P}_{1,x} \geq \mathbf{P}_{2,x}, \mathbf{P}_{1,y}, \mathbf{P}_{2,y}$ *since one could always permute the matrix to obtain such a structure.*

Since the return and number of iterations for the same policy is independent of the user index, we drop the subscript t in the rest of this subsection and use .

	Type x	Type y
Category 1	$\mathbf{P}_{1,x}$	$\mathbf{P}_{1,y}$
Category 2	$\mathbf{P}_{2,x}$	$\mathbf{P}_{2,y}$
Prior	$\mathbf{q}_x$	$\mathbf{q}_y = 1 - \mathbf{q}_x$

Table 3.2: Click probabilities for two user types and two categories.

As is well-known in the POMDP literature [74], the optimal policy π^* and its expected return are functions of belief states that represent the probability of the state at each

time. In our setting, the states are the user types. We denote by b_j the belief that the state is (type) x at iteration j. Similarly, $1 - b_j$ is the belief that the state is (type) y at iteration j. Needless to say, once the state $\perp$ is reached, the belief over the type states $[M]$ is irrelevant, as users do not come back. Nevertheless, we neglect this case as our analysis does not make use it.

We now describe how to compute the belief. At iteration $j = 1$, the belief state is set to be $b_1 = \mathbb{P}(state = x) = \mathbf{q}_x$. At iteration $j > 1$, upon receiving a positive reward $r_j = 1$, the belief is updated from $b_{j-1} \in [0, 1]$ to

$$b_j(b_{j-1}, a, 1) = \frac{b_{j-1} \cdot \mathbf{P}_{a,x}}{b_{j-1} \cdot \mathbf{P}_{a,x} + \mathbf{P}_{a,y}(1 - b_{j-1})}, \tag{3.15}$$

where we note that in the event of no-click, the current user departs the system, i.e., we move to the absorbing state $\perp$. For any policy $\pi : [0, 1] \to \{1, 2\}$ that maps a belief to a category, its expected return satisfies the Bellman equation,

$$\mathbb{E}[V^\pi(b)] = \Big(b\mathbf{P}_{\pi(b),x} + (1 - b)\mathbf{P}_{\pi(b),y}\Big) \cdot (1 + \mathbb{E}[V^\pi(b'(b, \pi(b), 1))]).$$

To better characterize the expected return, we introduce the following notion of belief-category walk.

Definition 3.41 (Belief-category walk)**.** *Let $\pi : [0, 1] \to \{1, 2\}$ be any policy. The sequence*

$$b_1, a_1 = \pi(b_1), b_2, a_2 = \pi(b_2), \ldots$$

is called the belief-category walk. Namely, it is the induced walk of belief updates and categories chosen by π, given all the rewards are positive ($r_j = 1$ for every $j \in \mathbb{N}$).

Notice that every policy induces a single, well-defined and deterministic belief-

category walk (recall that we assume departure-probabilities satisfy $\mathbf{\Lambda}_{a,\tau} = 1$ for every $a \in [K], \tau \in [M]$). Moreover, given any policy π, the trajectory of every user recommended by π is fully characterized by belief-category walk clipped at $b_{N^\pi(t)}, a_{N^\pi(t)}$.

In what follows, we derive a closed-form expression for the expected return as a function of b, the categories chosen by the policy, and the click-probability matrix.

Theorem 3.42. *For every policy π and an initial belief $b \in [0, 1]$, the expected return is given by*

$$\mathbb{E}[V^\pi(b)] = \sum_{i=1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}},$$

where $m_{1,i} := |\{a_j = 1, j \leq i\}|$ and $m_{2,i} := |\{a_j = 2, j \leq i\}|$ are calculated based on the belief-category walk $b_1, a_1, b_2, a_2, \ldots$ induced by π.

Characterizing the Optimal Policy

Using Theorem 3.42, we show that the planning problem can be solved in $O(1)$. To arrive at this conclusion, we perform a case analysis over the following three structures of the click-probability matrix $\mathbf{P}$:

- *Dominant Row*, where $\mathbf{P}_{1,y} \geq \mathbf{P}_{2,y}$;
- *Dominant Column*, where $\mathbf{P}_{2,x} \geq \mathbf{P}_{2,y} > \mathbf{P}_{1,y}$;
- *Dominant Diagonal*, where $\mathbf{P}_{1,x} \geq \mathbf{P}_{2,y} > \mathbf{P}_{1,y}, \mathbf{P}_{2,x}$.

Crucially, any matrix $\mathbf{P}$ takes exactly one of the three structures. Further, since $\mathbf{P}$ is known in the planning problem, identifying the structure at hand takes $O(1)$ time. Using this structure partition, we characterize the optimal policy.

Dominant Row We start by considering the simplest structure, in which the Category 1 is preferred by both types of users: Since $\mathbf{P}_{1,y} \geq \mathbf{P}_{2,y}$ and $\mathbf{P}_{1,x} \geq \mathbf{P}_{2,x}, \mathbf{P}_{1,y}, \mathbf{P}_{2,y}$ (Remark 3.40), there exists a dominant row, i.e., Category 1.

Lemma 3.43. *For any instance such that* $\mathbf{P}$ *has a dominant row* a*, the fixed policy* π^a *is an optimal policy.*

As expected, if Category 1 is dominant then the policy that always recommends Category 1 is optimal.

Dominant Column In the second structure we consider the case where there is no dominant row, and that the column of type x is dominant, i.e., $\mathbf{P}_{1,x} \geq \mathbf{P}_{2,x} \geq \mathbf{P}_{2,y} > \mathbf{P}_{1,y}$. In such a case, which is also the one described in the example in Section 3.2.2, it is unclear what the optimal policy would be since none of the categories dominates the other.

Surprisingly, we show that the optimal policy can be of only one form: Recommend Category 2 for some time steps (possibly zero) and then always recommend Category 1. To identify when to switch from Category 2 to Category 1, one only needs to compare four expected returns.

Theorem 3.44. *For any instance such that* $\mathbf{P}$ *has a dominant column, one of the following four policies is optimal:*

$$\pi^1, \pi^2, \pi^{2:\lfloor N^* \rfloor}, \pi^{2:\lceil N^* \rceil},$$

where $N^* = N^*(\mathbf{P}, \mathbf{q})$ *is a constant, and* $\pi^{2:\lfloor N^* \rfloor}$ *(*$\pi^{2:\lceil N^* \rceil}$*) stands for recommending Category* 2 *until iteration* $\lfloor N^* \rfloor$ *(*$\lceil N^* \rceil$*) and then switching to Category* 1*.*

The intuition behind the theorem is as follows. If the prior tends towards type y, we might start with recommending Category 2 (which users of type y are more likely to

click on). But after several iterations, and as long as the user stays, the posterior belief b increases since $\mathbf{P}_{2,x} > \mathbf{P}_{2,y}$ (recall Equation (3.15)). Consequently, since type x becomes more probable, and since $\mathbf{P}_{1,x} \geq \mathbf{P}_{2,x}$, the optimal policy recommends the best category for this type, i.e., Category 1. For the exact expression of N^*, we refer the reader to Section 3.2.10.

Using Theorem 3.42, we can compute the expected return for each of the four policies in $O(1)$, showing that we can find the optimal policy when $\mathbf{P}$ has a column in $O(1)$.

Dominant Diagonal In the last structure, we consider the case where there is no dominant row (i.e., $\mathbf{P}_{2,y} > \mathbf{P}_{1,y}$) nor a dominant column (i.e., $\mathbf{P}_{2,y} > \mathbf{P}_{2,x}$). At first glance, this case is more complex than the previous two, since none of the categories and none of the types dominates the other one. However, we uncover that the optimal policy can be either always recommending Category 1 or always recommending Category 2. Theorem 3.45 summarizes this result.

Theorem 3.45. *For any instance such that* $\mathbf{P}$ *has a dominant diagonal, either* π^1 *or* π^2 *is optimal.*

With the full characterization of the optimal policy derived in this section (for all the three structures), we have shown that the optimal policy can be computed in $O(1)$.

Learning: UCB-based Regret Bound

In this section, we move from the planning task to the learning one. Building on the results of previous sections, we know that there must exist a threshold policy—a policy whose belief-category walk has a finite prefix of one category, and an infinite suffix with the other category—which is optimal. However, there can still be infinitely

many such policies. To address this problem, we first show how to reduce the search space for approximately optimal policies with negligible additive factor to a set of $|\Pi| = O(\ln(T))$ policies. Then, we derive the parameters $\tilde{\tau}$ and η required for Algorithm 11. As an immediate consequence, we get a sublinear regret algorithm for this setting. We begin with defining threshold policies.

Definition 3.46 (Threshold Policy)**.** *A policy π is called an (a, h)-threshold policy if there exists an number $h \in \mathbb{N} \cup \{0\}$ in π's belief-category walk such that*

- *π recommends category a in iterations $j \leq h$, and*
- *π recommends category a' in iterations $j > h$,*

for $a, a' \in \{1, 2\}$ and $a \neq a'$.

For instance, the policy π^1 that always recommends Category 1 is the $(2, 0)$-threshold policy, as it recommends Category 2 until the zero'th iteration (i.e., never recommends Category 2) and then Category 1 eternally. Furthermore, the policy $\pi^{2:\lfloor N^* \rfloor}$ introduced in Theorem 3.44 is the $(2, \lfloor N^* \rfloor)$-threshold policy.

Next, recall that the chance of departure in every iteration is greater or equal to ϵ, since we assume $\max_{a,\tau} \mathbf{P}_{a,\tau} \leq 1 - \epsilon$. Consequently, the probability that a user will stay beyond H iterations is exponentially decreasing with H. We could use high-probability arguments to claim that it suffices to focus on the first H iterations, but without further insights this would yield $\Omega(2^H)$ candidates for the optimal policy. Instead, we exploit our insights about threshold policies.

Let Π_H be the set of all (a, h)-threshold policies for $a \in \{1, 2\}$ and $h \in [H] \cup \{0\}$. Clearly, $|\Pi_H| = 2H + 2$. Lemma 3.47 shows that the return obtained by the best policy in Π_H is not worse than that of the optimal policy π^* by a negligible factor.

Lemma 3.47. *For every $H \in \mathbb{N}$, it holds that*

$$\mathbb{E}\left[V^{\pi^*} - \max_{\pi \in \Pi_H} V^{\pi}\right] \leq \frac{1}{2^{O(H)}}.$$

Before we describe how to apply Algorithm 11, we need to show that returns of all the policies in Π_H are sub-exponential. In Lemma 3.48, we show that V^{π} is (τ^2, b)-sub-exponential for every threshold policy $\pi \in \Pi_H$, and provide bounds for both τ and b^2/τ^2.

Lemma 3.48. *Let $\tilde{\tau} = \frac{8e}{\ln(\frac{1}{1-\epsilon})}$ and $\eta = 1$. For every threshold policy $\pi \in \Pi_H$, the centred random variable $V^{\pi} - \mathbb{E}[V^{\pi}]$ is (τ^2, b)-sub-exponential with (τ^2, b) satisfying $\tilde{\tau} \geq \tau$ and $\eta \geq b^2/\tau^2$.*

We are ready to wrap up our solution for the learning task proposed in this section. Let $H = \Theta(\ln T)$, Π_H be the set of threshold policies characterized before, and let $\tilde{\tau}$ and η be constants as defined in Lemma 3.48.

Theorem 3.49. *Applying Algorithm 11 with $\Pi_H, T, \tilde{\tau}, \eta$ on the class of two-types two-categories instances considered in this section always yields an expected regret of*

$$\mathbb{E}[R_T] \leq O(\sqrt{T} \ln T).$$

Proof. It holds that

$$\begin{aligned}
\mathbb{E}[R_T] &= \mathbb{E}\left[TV^{\pi^*} - \sum_{t=1}^{T} V^{\pi_t}\right] \\
&= \mathbb{E}\left[TV^{\pi^*} - \max_{\pi \in \Pi_H} TV^{\pi}\right] + \mathbb{E}\left[\max_{\pi \in \Pi_H} TV^{\pi} - \sum_{t=1}^{T} V^{\pi_t}\right] \\
&\leq \frac{T}{2^{O(H)}} + O(\sqrt{HT \log T} + H \log T) = O(\sqrt{T} \ln T),
\end{aligned}$$

where the inequality follows from Theorem 3.35 and Lemma 3.47. Finally, setting $H = \Theta(\ln T)$ yields the desired result. ■

3.2.6 Extension: Planning Beyond Two User Types

In this section, we treat the planning task with two categories ($K = 2$) but potentially many types (i.e., $M \geq 2$). For convenience, we formalize the results in this section in terms of $M = 2$, but the results are readily extendable for the more general $2 \times M$ case. We derive an almost-optimal planning policy via dynamic programming, and then explain why it cannot be used for learning as we did in the previous section.

For reasons that will become apparent later on, we define by V_H^π as the return of a policy π until the H's iteration. Using Theorem 3.42, we have that

$$\mathbb{E}[V_H^\pi(b)] = \sum_{i=1}^{H} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}},$$

where $m_{1,i} := |\{a_j = 1, j \leq i\}|$ and $m_{2,i} := |\{a_j = 2, j \leq i\}|$ are calculated based on the belief-category walk $b_1, a_1, b_2, a_2, \ldots$ induced by π. Further, let $\tilde{\pi}^*$ denote the policy maximizing V_H.

Notice that there is a bijection from $H-$iterations policies to $(m_{1,i}, m_{2,i})_{i=1}^H$; hence, we can find $\tilde{\pi}^*$ by finding the arg max of the expression on the right-hand-side of the above equation, in terms of $(m_{1,i}, m_{2,i})_{i=1}^H$. Formally, we want to solve the integer linear programming (ILP),

$$\begin{aligned} \text{maximize } & \sum_{i=1}^{H} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}} \\ \text{subject to } & m_{a,i} = \sum_{l=1}^{i} z_{a,l} \text{ for } a \in \{1,2\}, i \in [H], \\ & z_{a,i} \in \{0,1\} \text{ for } a \in \{1,2\}, i \in [H], \\ & z_{1,i} + z_{2,i} = 1 \text{ for } i \in [H]. \end{aligned} \tag{3.16}$$

Despite that this problem involves integer programming, we can solve it using dynamic

programming in $O(H^2)$ runtime. Notice that the optimization is over a subset of binary variables $(z_{1,i}, z_{2,i})_{i=1}^{H}$. Let Z^H be the set of feasible solutions of the ILP, and similarly let Z^h denote set of prefixes of length $h \leq H$ of Z^H.

For any $h \in [H]$ and $\mathbf{z} \in Z^h$, define

$$D^h(\mathbf{z}) \stackrel{\text{def}}{=} \sum_{i=1}^{h} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}},$$

where $m_{a,i} = \sum_{l=1}^{i} z_{a,l}$ for $j \in \{1,2\}, i \in [h]$ as in the ILP.

Consequently, solving the ILP is equivalent to maximizing D^H over the domain Z^H.

Next, for any $h \in [H]$ and two integers c_1, c_2 such that $c_1 + c_2 = h$, define

$$\tilde{D}^h(c_1, c_2) \stackrel{\text{def}}{=} \max_{\substack{\mathbf{z} \in Z^h, \\ m_{1,h}(\mathbf{z}) = c_1 \\ m_{2,h}(\mathbf{z}) = c_2}} D^h(\mathbf{z}). \tag{3.17}$$

Under this construction, $\max_{c_1,c_2} \tilde{D}^H(c_1, c_2)$ over c_1, c_2 such that $c_1 + c_2 = H$ is precisely the value of the ILP.

Reformulating Equation (3.17) for $h > 1$,

$$\tilde{D}^h(c_1,c_2) = \max_{\substack{z_1,z_2 \in \{0,1\} \\ z_1+z_2=1}} \left\{ \tilde{D}^{h-1}(c_1 - z_1, c_2 - z_2) + \alpha(c_1,c_2) \right\},$$

where $\alpha(m_1, m_2) \stackrel{\text{def}}{=} b \cdot x_1^{m_1} \cdot x_2^{m_2} + (1-b) y_1^{m_1} \cdot y_2^{m_2}$. For every h, there are only $h+1$ possible values $\tilde{D}^h$ can take: All the ways of dividing h into non-negative integers c_1 and c_2; therefore, having computed $\tilde{D}^{h-1}$ for all h feasible inputs, we can compute $\tilde{D}^h(c_1, c_2)$ in $O(h)$. Consequently, computing $\max_{c_1,c_2} \tilde{D}^H(c_1, c_2)$, which is precisely the value of the ILP in (3.16), takes $O(H^2)$ run-time. Moreover, the policy $\tilde{\pi}^*$ can be found using backtracking. We remark that an argument similar to Lemma 3.47

implies that $\mathbb{E}[V^{\pi^*} - V^{\tilde{\pi}^*}] \leq \frac{1}{2^{O(H)}}$; hence, $\tilde{\pi}^*$ is almost optimal.

To finalize this section, we remark that this approach could also work for $K > 2$ categories. Naively, for a finite horizon H, there are K^H possible policies. The dynamic programming procedure explain above makes the search operate in run-time of $O(H^K)$. The run-time, exponential in the number of categories but polynomial in the horizon, is feasible when the number of categories is small.

3.2.7 Extension: How to Evaluate Experimentally

For general real-world datasets, we propose a scheme to construct semi-synthetic problem instances with many arms and many user types, using rating data sets with multiple ratings per user. We exemplify our scheme on the MovieLens Dataset [64]. As a pre-processing step, we set movie genres to be the categories of interest, select a subset of categories $|A|$ of size k (e.g., sci-fi, drama, and comedy), and select the number of user types, m. Remove any user who has not provided a rating for at least one movie from each category $a \in A$. When running the algorithm, randomly draw users from the data, and given a recommended category a, suggest them a random movie which they have rated, and set their click probability to $1 - r$, where $r \in [0, 1]$ is their normalized rating of the suggested movie.

3.2.8 Bernstein's Inequality

An important tool for analyzing sub-exponential random variables is Bernstein's Inequality, which is a concentration inequality for sub-exponential random variables (see, e.g., [72]). Being a major component of the regret analysis for Algorithm 11, we state it here for convenience.

Lemma 3.50. *(Bernstein's Inequality) Let a random variable X be sub-exponential*

with parameters (τ^2, b). *Then for every* $v \geq 0$*:*

$$\Pr[|X - \mathbb{E}[X]| \geq v] \leq \begin{cases} 2\exp(-\frac{v^2}{2\tau^2}) & v \leq \frac{\tau^2}{b} \\ 2\exp(-\frac{v}{2b}) & else \end{cases}.$$

3.2.9 Proofs for Single User Type (Section 3.2.4)

To simplify the proofs, we use the following notation: For a fixed-arm policy π^a, we use $V_j^{\pi^a}$ to denote its return from iteration j until the user departs. Namely,

$$V_j^{\pi^a} = \sum_{i=j}^{N^{\pi^a}} \mathbf{P}_a$$

Throughout this section, we will use the following Observation.

Observation 3.51. *For every policy* π *and iteration* j*,*

$$\mathbb{E}[V_j^\pi] = \mathbf{P}_{\pi_j}(1+\mathbb{E}[V_{j+1}^\pi])+(1-\mathbf{\Lambda}_{\pi_j})(1-\mathbf{P}_{\pi_j})\mathbb{E}[V_{j+1}^\pi] = \mathbb{E}[V_{j+1}^\pi](1-\mathbf{\Lambda}_{\pi_j}(1-\mathbf{P}_{\pi_j}))+\mathbf{P}_{\pi_j}.$$

Lemma 3.36. *A policy* π^{a^*} *is optimal if*

$$a^* \in \operatorname*{argmax}_{a\in[K]} \frac{\mathbf{P}_a}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)}.$$

Proof. First, recall that every POMDP has an optimal Markovian policy which is deterministic (we refer the reader to Section 3.2.5 for full formulation of the problem as POMDP). Having independent rewards and a single state implies that there exists $\mu^* \in \mathbb{N}$ such that $\mathbb{E}[V_j^*] = \mu^*$ for every $j \in \mathbb{N}$ (similarly to standard MAB problems, there exists a fixed-arm policy which is optimal).

Assume by contradiction that the optimal policy π^{a^*} holds

$$a^* \notin \operatorname*{argmax}_{a\in[k]} \frac{\mathbf{P}_a}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)}.$$

Now, notice that

$$\mathbb{E}[V^{\pi^{a'}}] = \mathbb{E}[V_1^{\pi^{a'}}] = \mathbb{E}[V_2^{\pi^{a'}}](1-\mathbf{\Lambda}_{a'}(1-\mathbf{P}_{a'})) + \mathbf{P}_{a'}$$

Solving the recurrence relation and summing the geometric series we get

$$\mathbb{E}[V^{\pi^{a'}}] = \mathbf{P}_{a'}\sum_{j=0}^{\infty}(1-\mathbf{\Lambda}_{a'}(1-\mathbf{P}_{a'}))^j = \frac{\mathbf{P}_{a'}}{\mathbf{\Lambda}_{a'}(1-\mathbf{P}_{a'})}.$$

Finally,

$$a^* \notin \operatorname*{argmax}_{a\in[k]} \frac{\mathbf{P}_a}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)},$$

yields that any fixed-armed policy, $\pi^{a'}$ such that

$$a' \in \operatorname*{argmax}_{a\in[k]} \frac{\mathbf{P}_a}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)}$$

holds $\mathbb{E}[V^{\pi^{a'}}] > \mathbb{E}[V^{\pi^{a^*}}]$, a contradiction to the optimality of π^{a^*} . ∎

Lemma 3.52. *For each $a \in [k]$, the centered random return $V^{\pi^a} - \mathbb{E}[V^{\pi^a}]$ is sub-exponential with parameter $C_2 = -4/\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))$.*

In order to show that returns of fixed-arm policies are sub-exponential random variables, we first show that the number of iterations of users recommended by fixed-arm policies is also a sub-exponential. For this purpose, we state here a lemma that implies that every geometric r.v. is a sub-exponential r.v.. The proof of the next lemma appears, e.g., in [68] (Lemma 4.3).

Lemma 3.53. *Let X be a geometric random variable with parameter $r \in (0,1)$, so that:*

$$\Pr[X = x] = (1-r)^{x-1} r, \quad x \in \mathbb{N}.$$

Then X satisfies Property (2) from Definition 3.34. Namely, X is sub-exponential with parameter $C_2 = -2/\ln(1-r)$. Formally,

$$\forall p \geq 0 : (\mathbb{E}[|X|^p])^{1/p} \leq -\frac{2}{\ln(1-r)} p.$$

The lemma above and Observation 3.37 allow us to deduce that the variables N_a are sub-exponential in the first part of the following Corollary (the case in which $\mathbf{\Lambda}_a = 0$ follows immediately from definition.). The second part of the lemma follows directly from the equivalences between Properties (2) and (1) in Definition 3.34.

Corollary 3.54. *For each $a \in [K]$, the number of iterations a user recommended by π^a stays within the system, N_a, is sub-exponential with parameter $C_2^a = -2/\ln(1 - \mathbf{\Lambda}_a(1-\mathbf{P}_a))$. In addition, there exist constants $C_1^a > 0$ for every $a \in [K]$ such that*

$$\forall a \in [K],\ v \geq 0 : \Pr[|N_a| > v] \leq \exp(1 - \frac{v}{C_1^a}).$$

The next Proposition 3.55 is used for the proof of Lemma 3.52.

Proposition 3.55. *For every $a \in [K]$,*

$$|\mathbb{E}[V^{\pi^a}]| \leq \frac{-2}{\ln(1 - \mathbf{\Lambda}_a(1-\mathbf{P}_a))}$$

Proof. First, notice that

$$(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a)) > (1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))\frac{-\mathbf{\Lambda}_a(1-\mathbf{P}_a)}{1-\mathbf{\Lambda}_a(1-\mathbf{P}_a)} = -\mathbf{\Lambda}_a(1-\mathbf{P}_a) > -2\mathbf{\Lambda}_a(1-\mathbf{P}_a),$$

where the first inequality is due to $\frac{x}{1+x} \leq \ln(1+x)$ for every $x \geq -1$. Rearranging,

$$\frac{1-\mathbf{\Lambda}_a(1-\mathbf{P}_a)}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)} < \frac{-2}{\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))}. \tag{3.18}$$

For each user, the realization of V^{π^a} is less or equal to the realization of $N_a - 1$ for the same user (as users provide negative feedback in their last iteration); hence,

$$|\mathbb{E}[V^{\pi^a}]| = \mathbb{E}[V^{\pi^a}] \leq \mathbb{E}[N_a] - 1 = \frac{1}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)} - 1 = \frac{1-\mathbf{\Lambda}_a(1-\mathbf{P}_a)}{\mathbf{\Lambda}_a(1-\mathbf{P}_a)} < \frac{-2}{\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))}.$$

■

We proceed by showing that returns of fix-armed policies satisfy Property (1) from Definition 3.34.

Lemma 3.52. *For each $a \in [k]$, the centered random return $V^{\pi^a} - \mathbb{E}[V^{\pi^a}]$ is sub-exponential with parameter $C_2 = -4/\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))$.*

Proof. We use Property (1) from Definition 3.34 to derive that V^{π^a} is also sub-exponential. This is true since the tails of V^{π^a} satisfy that for all $v \geq 0$,

$$\Pr[|V^{\pi^a}| > v] \leq \Pr[|N_a| > v+1] \leq \Pr[|N_a| > v] \leq_{(1)} \exp(1 - \frac{v}{C_1}),$$

where the first inequality follows since $|N_a| > v+1$ is a necessary condition for $|V^{\pi^a}| > v$, and the last inequality follows from Corollary 3.54. Along with Definition 3.34, we conclude that

$$\mathbb{E}[|V^{\pi^a}|^p]^{1/p} \leq -2/\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))p. \tag{3.19}$$

Now, applying Minkowski's inequality and then Jensen's inequality (as $f(z) =$

$z^p, g(z) = |z|$ are convex for every $p \geq 1$) we get

$$(\mathbb{E}[|V^{\pi^a} - \mathbb{E}[V^{\pi^a}]|^p])^{1/p} \leq \mathbb{E}[|V^{\pi^a}|^p]^{1/p} + \mathbb{E}[\mathbb{E}[|V^{\pi^a}|]^p]^{1/p} \leq \mathbb{E}[|V^{\pi^a}|^p]^{1/p} + |\mathbb{E}[V^{\pi^a}]|.$$

Using Proposition 3.55 and Inequality (3.19), we get

$$\mathbb{E}[|V^{\pi^a}|^p]^{1/p} + |\mathbb{E}[V^{\pi^a}]| \leq \frac{-2}{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))} + \frac{1}{\mathbf{\Lambda}_a(1 - \mathbf{P}_a)} - 1 \leq \frac{-4}{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))}$$

Hence $V^{\pi^a} - \mathbb{E}[V^{\pi^a}]$ is sub-exponential with parameter $C_2 = -4/\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))$. ■

Lemma 3.38. *For each category $a \in [K]$, the centred random variable $V^{\pi^a} - \mathbb{E}[V^{\pi^a}]$ is sub-exponential with parameters (τ_a^2, b_a), such that*

$$\tau_a = b_a = -\frac{8e}{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))}.$$

Proof. Throughout this proof, we will use the sub-exponential norm, $||\cdot||_{\psi_1}$, which is defined as

$$||Z||_{\psi_1} = \sup_{p \geq 1} \frac{(\mathbb{E}[|Z|^p])^{1/p}}{p}.$$

Let

$$X = \frac{V^{\pi^a} - \mathbb{E}[V^{\pi^a}]}{||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1}}, \quad y = \gamma \cdot ||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1}.$$

We have that

$$||X||_{\psi_1} = ||\frac{V^{\pi^a} - \mathbb{E}[V^{\pi^a}]}{||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1}}||_{\psi_1} = 1. \tag{3.20}$$

Let γ be such that $|\gamma| < 1/b_a = -\frac{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))}{8e}$. From Lemma 3.52 we conclude that

$$|\gamma| = \Big|\frac{y}{||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1}}\Big| \leq -\frac{\ln(1 - \mathbf{\Lambda}_a(1 - \mathbf{P}_a))}{8e} = \frac{1}{2e} \cdot \frac{1}{||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1}};$$

hence, $|y| < \frac{1}{2e}$.

Summing the geometric series, we get

$$\sum_{p=2}^{\infty}(e|y|)^p = \frac{e^2y^2}{1-e|y|} < 2e^2y^2 \tag{3.21}$$

In addition, notice that $yX = \gamma(V^{\pi^a} - \mathbb{E}[V^{\pi^a}])$.

Next, from the Taylor series of $\exp(\cdot)$ we have

$$\mathbb{E}[\exp(\gamma(V^{\pi^a} - \mathbb{E}[V^{\pi^a}]))] = \mathbb{E}[\exp(yX)] = 1 + y\mathbb{E}[x] + \sum_{p=2}^{\infty}\frac{y^p\mathbb{E}[X^p]}{p!}.$$

Combining the fact that $\mathbb{E}[X] = 0$ and (3.20) to the above,

$$1 + y\mathbb{E}[x] + \sum_{p=2}^{\infty}\frac{y^p\mathbb{E}[X^p]}{p!} \leq 1 + \sum_{p=2}^{\infty}\frac{y^pp^p}{p!}.$$

By applying $p! \geq (\frac{p}{e})^p$ and (3.21), we get

$$1+\sum_{p=2}^{\infty}\frac{y^pp^p}{p!} \leq 1+\sum_{p=2}^{\infty}(e|y|)^p \leq 1+2e^2y^2 \leq \exp(2e^2y^2) = \exp(2e^2(\gamma\cdot||V^{\pi^a}-\mathbb{E}[V^{\pi^a}]||_{\psi_1})^2),$$

where the last inequality is due to $1 + x \leq e^x$.

Note that $||V^{\pi^a} - \mathbb{E}[V^{\pi^a}]||_{\psi_1} \leq -\frac{4}{\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))})^2)$. Ultimately,

$$\mathbb{E}[\exp(\gamma(V^{\pi^a}-\mathbb{E}[V^{\pi^a}]))] \leq \exp\left(2e^2\gamma^2(-\frac{4}{\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))})^2\right) = \exp\left(\frac{1}{2}\gamma^2(-\frac{8e}{\ln(1-\mathbf{\Lambda}_a(1-\mathbf{P}_a))})^2\right).$$

This concludes the proof of the lemma. ■

3.2.10 Proofs for Two User Types and Two Categories (Section 3.2.5)

Planning when $K = 2$

Theorem 3.42. *For every policy π and an initial belief $b \in [0, 1]$, the expected return is given by*

$$\mathbb{E}[V^{\pi}(b)] = \sum_{i=1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}},$$

where $m_{1,i} := |\{a_j = 1, j \leq i\}|$ and $m_{2,i} := |\{a_j = 2, j \leq i\}|$ are calculated based on the belief-category walk $b_1, a_1, b_2, a_2, \ldots$ induced by π.

Proof. Let $\beta_i^{\pi}(b) := b \cdot \mathbf{P}_{1,x}{}^{m_{1,i}} \cdot \mathbf{P}_{2,x}{}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}{}^{m_{1,i}} \cdot \mathbf{P}_{2,y}{}^{m_{2,i}}$. We will prove that for every policy π and every belief b, we have that $\mathbb{E}[V_H^{\pi_a}(b)] = \sum_{i=1}^{H} \beta_i^{\pi}(b)$ by a backward induction over H.

For the base case, consider $H = 1$. We have that

$$\mathbb{E}[V_1^{\pi}(b_1)] = b_1 \cdot \mathbf{P}_{a_1,x} + (1-b)\mathbf{P}_{a_1,y} = b \cdot \mathbf{P}_{1,x}{}^{m_{1,1}} \cdot \mathbf{P}_{2,x}{}^{m_{2,1}} + (1-b)\mathbf{P}_{1,y}{}^{m_{1,1}} \cdot \mathbf{P}_{2,y}{}^{m_{2,1}} = \beta_1^{\pi}(b)$$

as $m_{a,1} = \mathbb{I}[a_1 = a]$.

For the inductive step, assume that $\mathbb{E}[V_{H-1}^{\pi}(b)] = \sum_{i=1}^{H-1} \beta_i^{\pi}(b)$ for every $b \in [0, 1]$. We need to show that $\mathbb{E}[V_H^{\pi}(b)] = \sum_{i=1}^{H} \beta_i^{\pi}(b)$ for every $b \in [0, 1]$.

Indeed,

$$\begin{aligned}
\mathbb{E}[V_H^{\pi}(b_1)] &= \beta_1^{\pi}(b_1)(1 + \mathbb{E}[V_{H-1}^{\pi}(b'(b_1, a_1, liked))]) \\
&= \beta_1^{\pi}(b_1)(1 + \mathbb{E}[V_{H-1}^{\pi}(b_2)]) \\
&= \beta_1^{\pi}(b_1)(1 + \sum_{i=2}^{H-1} \beta_i^{\pi}(b_2)) \\
&= \sum_{i=1}^{H} \beta_i^{\pi}(b_1),
\end{aligned}$$

where the second to last equality is due to the induction hypothesis and the assumption that π is a deterministic stationary policy. The proof completes by realizing that $\mathbb{E}[V^{\pi}(b)] = \lim_{H\to\infty} \mathbb{E}[V_H^{\pi}(b)] = \lim_{H\to\infty} \sum_{i=1}^{H} \beta_i^{\pi}(b) = \sum_{i=1}^{\infty} \beta_i^{\pi}(b)$, since the sum is finite and has positive summands. ∎

Dominant Row (DR)

Lemma 3.43. *For any instance such that* $\mathbf{P}$ *has a dominant row* a*, the fixed policy* π^a *is an optimal policy.*

Proof. We will show that for every iteration j, no matter what were the previous topic recommendations were, selecting topic 1 rather than topic 2 can only increase the value.

Let π be a stationary policy such that $\pi(b_j) = 2$. Changing it into a policy π^j that is equivalent to π for all iterations but iteration $j+1$ in which it recommends topic 1 can only improve the value.

Since $\mathbf{P}_{1,x}, \mathbf{P}_{2,x}, \mathbf{P}_{1,y}, \mathbf{P}_{2,y} \geq 0$, $\mathbf{P}_{1,x} - \mathbf{P}_{2,x} \geq 0$, $b, 1-b \geq 0$ and this structure satisfies $\mathbf{P}_{2,y} - \mathbf{P}_{1,y} \leq 0$, we get that for every $\bar{m}_{1,j}, \bar{m}_{2,j}, n_{1,j}, n_{2,j} \in \mathbb{N}$ and for every b,

$$b \cdot \mathbf{P}_{1,x}^{\bar{m}_{1,j}+n_{1,j}} \cdot \mathbf{P}_{2,x}^{\bar{m}_{2,j}+n_{2,j}} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) \geq (1-b) \mathbf{P}_{1,y}^{\bar{m}_{1,j}+n_{1,j}} \cdot \mathbf{P}_{2,y}^{\bar{m}_{2,j}+n_{2,j}} (\mathbf{P}_{2,y} - \mathbf{P}_{1,y});$$

thus,

$$b \cdot \mathbf{P}_{1,x}^{\bar{m}_{1,j}+1+n_{1,j}} \cdot \mathbf{P}_{2,x}^{\bar{m}_{2,j}+n_{2,j}} + (1-b) \mathbf{P}_{1,y}^{\bar{m}_{1,j}+1+n_{1,j}} \cdot \mathbf{P}_{2,y}^{m_{2,j}+n_{2,j}} \geq$$

$$b \cdot \mathbf{P}_{1,x}^{\bar{m}_{1,j}+n_{1,j}} \cdot \mathbf{P}_{2,x}^{\bar{m}_{2,j}+1+n_{2,j}} + (1-b) \mathbf{P}_{1,y}^{\bar{m}_{1,j}+n_{1,j}} \cdot \mathbf{P}_{2,y}^{\bar{m}_{2,j}+1+n_{2,j}}.$$

Hence for every time step $j+1$, choosing topic 1 instead of topic 2 leads to increased value of each of the summation element $b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b) \mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}}$ such

that $m_{1,i} = \bar{m}_{1,j} + n_{1,j} \geq \bar{m}_{1,j}$ and $m_{2,i} = \bar{m}_{2,j} + n_{2,j} \geq \bar{m}_{2,j}$. We deduce that

$$\mathbb{E}[V^{\pi^j}(b)] \geq \mathbb{E}[V^{\pi}(b)].$$

■

Dominant Column (DC)

Before proving the main theorem (Theorem 3.44), we prove two auxiliary lemmas.

Lemma 3.56. *For* $\mathbf{P}$ *with a DC structure, if a policy* π *is optimal then it recommends topic* 1 *for all iteration* $j' \geq j+1$ *such that*

$$\sum_{i=j+1}^{\infty} \mathbf{P}_{1,x}^{m_{1,i}} \mathbf{P}_{2,x}^{m_{2,i}} > \quad \sum_{i=j+1}^{\infty} \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \mathbf{P}_{1,y}^{m_{1,i}} \mathbf{P}_{2,y}^{m_{2,i}}. \tag{3.22}$$

Proof. First, assume by contradiction that there exists an optimal policy π that recommends topic 2 in iteration $j+1$ such that (3.22) holds.

Let π^j be the policy that is equivalent to π but recommend topic 1 instead of topic 2 in iteration $j+1$. Since π and π^j recommends the same topic until iteration j, along with the optimality of π, we have

$$\mathbb{E}[V^{\pi^j}(b)] - \mathbb{E}[V^{\pi}(b)] = \mathbb{E}[V_{j+1}^{\pi^j}(b)] - \mathbb{E}[V_{j+1}^{\pi}(b)] \leq 0.$$

Expending the above equation,

$$\sum_{i=j+1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m_{1,i}^{\pi}+1} \cdot \mathbf{P}_{2,x}^{m_{2,i}^{\pi}-1} + (1-b) \mathbf{P}_{1,y}^{m_{1,i}^{\pi}+1} \cdot \mathbf{P}_{2,y}^{m_{2,i}^{\pi}-1} - \Big(\sum_{i=j+1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m_{1,i}^{\pi}} \cdot \mathbf{P}_{2,x}^{m_{2,i}^{\pi}} + (1-b) \mathbf{P}_{1,y}^{m_{1,i}^{\pi}} \cdot \mathbf{P}_{2,y}^{m_{2,i}^{\pi}} \Big) \leq 0$$

$$\sum_{i=j+1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} (\frac{\mathbf{P}_{1,x}}{\mathbf{P}_{2,x}} - 1) \leq \sum_{i=j+1}^{\infty} (1-b) \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,y}^{m^{\pi}_{2,i}} (1 - \frac{\mathbf{P}_{1,y}}{\mathbf{P}_{2,y}})$$

$$\frac{b(\mathbf{P}_{1,x} - \mathbf{P}_{2,x})}{\mathbf{P}_{2,x}} \sum_{i=j+1}^{\infty} \cdot \mathbf{P}_{1,x}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} \leq \frac{(1-b)(\mathbf{P}_{2,y} - \mathbf{P}_{1,y})}{\mathbf{P}_{2,y}} \sum_{i=j+1}^{\infty} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,y}^{m^{\pi}_{2,i}}$$

$$\sum_{i=j+1}^{\infty} \mathbf{P}_{1,x}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} \leq \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \sum_{i=j+1}^{\infty} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \cdot \mathbf{P}_{2,y}^{m^{\pi}_{2,i}},$$

which is a contradiction to (3.22).

For the second part of the lemma, assume that condition (3.22) holds for some iteration $j+1 \in \mathbb{N}$ and some optimal policy π; hence, $\pi(b, m^{\pi}_{1,j}, m^{\pi}_{2,j}) = 1$ and we have $m^{\pi}_{1,j+1} = m^{\pi}_{1,j} + 1$ and $m^{\pi}_{2,j+1} = m^{\pi}_{2,j}$. Exploiting this fact, we have that

$$\sum_{i=j+2}^{\infty} \mathbf{P}_{1,x}^{m^{\pi}_{1,i}} \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} = \sum_{i=j+1}^{\infty} \mathbf{P}_{1,x}^{m^{\pi}_{1,i}+1} \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} = \mathbf{P}_{1,x} \sum_{i=j+1}^{\infty} \mathbf{P}_{1,x}^{m^{\pi}_{1,i}} \mathbf{P}_{2,x}^{m^{\pi}_{2,i}} > (3.22),$$

implying

$$\begin{aligned}
&\mathbf{P}_{1,x} \sum_{i=j+1}^{\infty} \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \mathbf{P}_{2,y}^{m^{\pi}_{2,i}} \\
>&(\mathbf{P}_{1,x} \geq \mathbf{P}_{1,y}) \mathbf{P}_{1,y} \sum_{i=j+1}^{\infty} \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \mathbf{P}_{2,y}^{m^{\pi}_{2,i}} \\
=&\sum_{i=j+1}^{\infty} \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}+1} \mathbf{P}_{2,y}^{m^{\pi}_{2,i}} \\
=&\sum_{i=j+2}^{\infty} \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \mathbf{P}_{1,y}^{m^{\pi}_{1,i}} \mathbf{P}_{2,y}^{m^{\pi}_{2,i}}.
\end{aligned}$$

■

An immediate consequence of Lemma 3.56 is the following corollary.

Corollary 3.57. *For any DC-structured* $\mathbf{P}$ *and every belief* $b \in [0,1]$*, the optimal*

policy π first recommends topic 2 for at most

$$argmin_N \sum_{i=1}^{N} \mathbf{P}_{2,x}^{m_{2,i}^{\pi}} > \frac{1-b}{b} \cdot \frac{\mathbf{P}_{2,y} - \mathbf{P}_{1,y}}{\mathbf{P}_{1,x} - \mathbf{P}_{2,x}} \cdot \frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \sum_{i=1}^{N} \mathbf{P}_{2,y}^{m_{2,i}^{\pi}}$$

times, and then recommends topic 1 permanently. In addition, $N \in \mathbb{N}$ since $\mathbf{P}_{2,x} > \mathbf{P}_{2,y}$.

Theorem 3.44. *For any instance such that $\mathbf{P}$ has a dominant column, one of the following four policies is optimal:*

$$\pi^1, \pi^2, \pi^{2:\lfloor N^* \rfloor}, \pi^{2:\lceil N^* \rceil},$$

where $N^* = N^*(\mathbf{P}, \mathbf{q})$ is a constant, and $\pi^{2:\lfloor N^* \rfloor}$ ($\pi^{2:\lceil N^* \rceil}$) stands for recommending Category 2 until iteration $\lfloor N^* \rfloor$ ($\lceil N^* \rceil$) and then switching to Category 1.

Proof. Due to Theorem 3.42 and Corollary 3.57, we can write the expected value of a policy as a function of N when $\mathbf{P}$ has a DC structure:

$$\begin{aligned}
\mathbb{E}[V^{\pi_N}(b)] &= \sum_{i=1}^{\infty} b \cdot \mathbf{P}_{1,x}^{m_{1,i}} \cdot \mathbf{P}_{2,x}^{m_{2,i}} + (1-b)\mathbf{P}_{1,y}^{m_{1,i}} \cdot \mathbf{P}_{2,y}^{m_{2,i}} \\
&= \sum_{i=1}^{N} b \cdot \mathbf{P}_{2,x}^{i} + (1-b)\mathbf{P}_{2,y}^{i} + \sum_{i=N+1}^{\infty} b \cdot \mathbf{P}_{2,x}^{N} \cdot \mathbf{P}_{1,x}^{i-N} + (1-b)\mathbf{P}_{2,y}^{N} \cdot \mathbf{P}_{1,y}^{i-N} \\
&= b \cdot \frac{\mathbf{P}_{2,x}(\mathbf{P}_{2,x}^{N} - 1)}{\mathbf{P}_{2,x} - 1} + (1-b) \cdot \frac{\mathbf{P}_{2,y}(\mathbf{P}_{2,y}^{N} - 1)}{\mathbf{P}_{2,y} - 1} + b \cdot \mathbf{P}_{2,x}^{N} \cdot \sum_{i=1}^{\infty} \mathbf{P}_{1,x}^{i} + (1-b)\mathbf{P}_{2,y}^{N} \cdot \sum_{i=1}^{\infty} \mathbf{P}_{1,y}^{i} \\
&= b \cdot \frac{\mathbf{P}_{2,x}(\mathbf{P}_{2,x}^{N} - 1)}{\mathbf{P}_{2,x} - 1} + (1-b) \cdot \frac{\mathbf{P}_{2,y}(\mathbf{P}_{2,y}^{N} - 1)}{\mathbf{P}_{2,y} - 1} + b \cdot \mathbf{P}_{2,x}^{N} \cdot \frac{\mathbf{P}_{1,x}}{1 - \mathbf{P}_{1,x}} + (1-b)\mathbf{P}_{2,y}^{N} \frac{\mathbf{P}_{1,y}}{1 - \mathbf{P}_{1,y}} \\
&= \mathbf{P}_{2,x}^{N} \cdot b\Big(\frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,x} - 1} + \frac{\mathbf{P}_{1,x}}{1 - \mathbf{P}_{1,x}}\Big) + \mathbf{P}_{2,y}^{N}(1-b)\Big(\frac{\mathbf{P}_{2,y}}{\mathbf{P}_{2,y} - 1} + \frac{\mathbf{P}_{1,y}}{1 - \mathbf{P}_{1,y}}\Big) + \frac{b\mathbf{P}_{2,x}}{1 - \mathbf{P}_{2,x}} + \frac{(1-b)\mathbf{P}_{2}}{1 - \mathbf{P}_{2,y}}
\end{aligned} \tag{3.23}$$

Equation (3.23) could be cast as $c_1 \cdot \mathbf{P}_{2,x}^{N} + c_2\mathbf{P}_{2,y}^{N} + c_3(\mathbf{P}_{2,x}, \mathbf{P}_{2,y})$ for **positive** c_1,

negative c_2 and positive c_3. Let $f : \mathbb{R} \leftarrow \mathbb{R}$ be the continuous function such that $f(N) = c_1 \cdot \mathbf{P}_{2,x}^N + c_2 \mathbf{P}_{2,y}^N + c_3(\mathbf{P}_{2,x}, \mathbf{P}_{2,y})$.

We take the derivative w.r.t. N to find the saddle point of f:

$$\frac{d}{dN} f = c_1 \cdot \ln \mathbf{P}_{2,x} \cdot \mathbf{P}_{2,x}^N + c_2 \ln \mathbf{P}_{2,y} \cdot \mathbf{P}_{2,y}^N = 0,$$

which suggests the saddle point of f is

$$\tilde{N} = \frac{\ln\left(- \frac{c_2 \ln \mathbf{P}_{2,y}}{c_1 \ln \mathbf{P}_{2,x}} \right)}{\ln\left(\frac{\mathbf{P}_{2,x}}{\mathbf{P}_{2,y}} \right)}.$$

Next, set $N^* \stackrel{\text{def}}{=} \max\{0, \tilde{N}\}$. Since f has a single saddle point and for every $n \in \mathbb{N}$ it holds that $f(N) = \mathbb{E}[V^{\pi_N}(b)]$, to determine the optimal policy, one only needs to compare the value $\mathbb{E}[V^{\pi_N}(b)]$ at the boundary points ($N = 0, N = \infty$) and at the closest integers to the saddle point ($N = \lfloor N^* \rfloor, N = \lceil N^* \rceil$). ∎

Dominant Diagonal (SD)

Theorem 3.45. *For any instance such that* $\mathbf{P}$ *has a dominant diagonal, either* π^1 *or* π^2 *is optimal.*

Proof. We prove the following claim by a backward induction over the number of iterations remaining: For every $k = H - 1, \ldots 1$ it holds that for every policy π and belief b,

$$\mathbb{E}[V_k^{\pi}(b)] \leq \max\{\mathbb{E}[V_k^{\pi^1}(b)], \mathbb{E}[V_k^{\pi^2}(b)]\}.$$

First, we notice that when $k = H - 1$, the only possible policies are π^1 and π^2. For $k = H - 2$, we prove the statement by contradiction. There are only two ways to

selects topics when $k = H - 2$:

$$\pi'_{1:H} = (\pi_{1:H-2}, \underbrace{1}_{H-1}, \underbrace{2}_{H}) \quad \text{and} \quad \pi''_{1:H} = (\pi_{1:H-2}, \underbrace{2}_{H-1}, \underbrace{1}_{H}).$$

Let m_1 and m_2 denote the number of times π has played topic 1 and 2 till time $H - 2$, inclusive. Assume that the policy π' is optimal. In particular, it holds that $\mathbb{E}[V_k^{\pi_1}] \leq \mathbb{E}[V_k^{\pi'}]$ and $\mathbb{E}[V_k^{\pi_2}] \leq \mathbb{E}[V_k^{\pi'}]$. We get

$$b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}\mathbf{P}_{1,x}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) \leq \mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)\mathbf{P}_{1,y}(\mathbf{P}_{2,y} - \mathbf{P}_{1,y}), \tag{3.24}$$

and

$$\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(\mathbf{P}_{2,y} - \mathbf{P}_{1,y})(1 + \mathbf{P}_{2,y}) \leq b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x})(1 + \mathbf{P}_{2,x}). \tag{3.25}$$

As both sides of (3.24) and (3.25) are positive, the multiplication of their left hand sides is smaller than the multiplication of their right hand sides, i.e.,

$$\begin{aligned} &b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}\mathbf{P}_{1,x}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x})\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(\mathbf{P}_{2,y} - \mathbf{P}_{1,y})(1 + \mathbf{P}_{2,y}) \\ \leq\ &\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)\mathbf{P}_{1,y}(\mathbf{P}_{2,y} - \mathbf{P}_{1,y})b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x})(1 + \mathbf{P}_{2,x}) \end{aligned}$$

Dividing both sides by $b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x})\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(\mathbf{P}_{2,y} - \mathbf{P}_{1,y}) > 0$, we obtain

$$\mathbf{P}_{1,x}(1 + \mathbf{P}_{2,y}) \leq \mathbf{P}_{1,y}(1 + \mathbf{P}_{2,x}),$$

which is a contradiction as $\mathbf{P}_{1,x} > \mathbf{P}_{1,y}$ and $1 + \mathbf{P}_{2,y} > 1 + \mathbf{P}_{2,x}$.

Now, assume that the policy π'' is optimal. In particular, it holds that $\mathbb{E}[V_k^{\pi^1}] \leq \mathbb{E}[V_k^{\pi''}]$

and $\mathbb{E}[V_k^{\pi^2}] \leq \mathbb{E}[V_k^{\pi''}]$. We get

$$\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}b(\mathbf{P}_{1,x}-\mathbf{P}_{2,x})(1+\mathbf{P}_{1,x}) \leq \mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(1+\mathbf{P}_{1,y})(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}), \quad (3.26)$$

and

$$\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)\mathbf{P}_{2,y}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) \leq \mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}b\mathbf{P}_{2,x}(\mathbf{P}_{1,x}-\mathbf{P}_{2,x}). \quad (3.27)$$

As both sides of (3.26) and (3.27) are positive, the multiplication of their left hand sides is smaller than the multiplication of their right hand sides,

$$\begin{aligned}&\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}b(\mathbf{P}_{1,x}-\mathbf{P}_{2,x})(1+\mathbf{P}_{1,x})\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)\mathbf{P}_{2,y}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y})\\ \leq\ &\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(1+\mathbf{P}_{1,y})(\mathbf{P}_{2,y}-\mathbf{P}_{1,y})\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}b\mathbf{P}_{2,x}(\mathbf{P}_{1,x}-\mathbf{P}_{2,x}).\end{aligned}$$

Dividing both sides by $\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2}b(\mathbf{P}_{1,x}-\mathbf{P}_{2,x})\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2}(1-b)(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) > 0$, we obtain

$$\mathbf{P}_{2,y}(1+\mathbf{P}_{1,x}) \leq \mathbf{P}_{2,x}(1+\mathbf{P}_{1,y}),$$

which is again, a contradiction as $\mathbf{P}_{2,x} < \mathbf{P}_{2,y}$ and $1+\mathbf{P}_{1,y} < 1+\mathbf{P}_{1,x}$.

For $H \geq 3$, we prove the statement by contradiction. Suppose not, i.e., the optimal policy π switch recommended topic at least once. Let t denote the time step where π switch for the last time. We first consider the case where π has switched from topic 2 to topic 1 at time t. More specifically, we have

$$\pi_{1:H} = (\pi_{1:t-2}, \underbrace{2}_{\pi_{t-1}}, \underbrace{1}_{\pi_t}, \underbrace{1,\ldots,1}_{\pi_{t+1:H-1}}, \underbrace{1}_{\pi_H}).$$

Consider another policy $\tilde{\pi}$ (that behaves the same as π except at time step $t-1$)

defined as

$$\tilde{\pi}_{1:H} = (\pi_{1:t-2}, \underbrace{2}_{\pi_{t-1}}, \underbrace{2}_{\pi_t}, \underbrace{1, \ldots, 1}_{\pi_{t+1:H-1}}, \underbrace{1}_{\pi_H}).$$

Let m_1 and m_2 denote the number of times π has recommended topic 1 and 2 till (and include) time $t-1$. Since π is optimal, we have the difference between the value of π and $\tilde{\pi}$ to be non-negative, i.e.,

$$\mathbb{E}[V_H^{\pi}] - \mathbb{E}[V_H^{\tilde{\pi}}] = \sum_{i=1}^{H-t+1} b\mathbf{P}_{1,x}^{m_1+i-1}\mathbf{P}_{2,x}^{m_2+1}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+i-1}\mathbf{P}_{2,y}^{m_2+1}(\mathbf{P}_{1,y} - \mathbf{P}_{2,y}) \geq 0, \tag{3.28}$$

where the difference is induced by the discrepancy of the two policies from time step t to H. Consider another policy π' (that behaves the same as π except at time step H) defined as

$$\pi'_{1:H} = (\pi_{1:t-2}, \underbrace{2}_{\pi_{t-1}}, \underbrace{1}_{\pi_t}, \underbrace{1, \ldots, 1}_{\pi_{t+1:H-1}}, \underbrace{2}_{\pi_H}).$$

Since π is optimal, we have the difference between the value of π and π' to be non-negative, i.e.,

$$\mathbb{E}[V_H^{\pi}] > \mathbb{E}[V_H^{\pi'}] \Rightarrow b\mathbf{P}_{1,x}^{m_1+H-t}\mathbf{P}_{2,x}^{m_2}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) > (1-b)\mathbf{P}_{1,y}^{m_1+H-t}\mathbf{P}_{2,y}^{m_2}(\mathbf{P}_{2,y} - \mathbf{P}_{1,y}),$$

where the difference is induced by the discrepancy of the two policies from time step H. Multiplying both sides by $\mathbf{P}_{1,y} > 0$, we get

$$\mathbf{P}_{1,y}b\mathbf{P}_{1,x}^{m_1+H-t}\mathbf{P}_{2,x}^{m_2}(\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) > (1-b)\mathbf{P}_{1,y}^{m_1+H-t+1}\mathbf{P}_{2,y}^{m_2}(\mathbf{P}_{2,y} - \mathbf{P}_{1,y}).$$

Using $\frac{\mathbf{P}_{1,x}}{\mathbf{P}_{1,y}} > 1$, and $\mathbf{P}_{1,y} b \mathbf{P}_{1,x}^{m_1+H-t} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) > 0$,

$$b\mathbf{P}_{1,x}^{m_1+H-t+1} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) > (1-b)\mathbf{P}_{1,y}^{m_1+H-t+1} \mathbf{P}_{2,y}^{m_2} (\mathbf{P}_{2,y} - \mathbf{P}_{1,y});$$

hence,

$$b\mathbf{P}_{1,x}^{m_1+H-t+1} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+H-t+1} \mathbf{P}_{2,y}^{m_2} (\mathbf{P}_{1,y} - \mathbf{P}_{2,y}) \geq 0. \quad (3.29)$$

Next, we construct a new policy π_{new} that outperforms π. We let π^{new} to be the policy defined as below

$$\pi_{1:H}^{\text{new}} = (\pi_{1:t-2}, \underbrace{1}_{\pi_{t-1}}, \underbrace{1}_{\pi_t}, \underbrace{1, \ldots, 1}_{\pi_{t+1:H-1}}, \underbrace{1}_{\pi_H}).$$

The value difference between π^{new} and π (caused by the discrepancy of the two policies from time $t-1$ to H) is

$$\begin{aligned}
\mathbb{E}[V_H^{\pi^{\text{new}}}] - \mathbb{E}[V_H^{\pi}] = & \sum_{i=1}^{H-t+1} b\mathbf{P}_{1,x}^{m_1+i-1} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+i-1} \mathbf{P}_{2,y}^{m_2} (\mathbf{P}_{1,y} - \mathbf{P}_{2,y}) \\
& + b\mathbf{P}_{1,x}^{m_1+H-t+1} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+H-t+1} \mathbf{P}_{2,y}^{m_2} (\mathbf{P}_{1,y} - \mathbf{P}_{2,y}) \\
& > \sum_{i=1}^{H-t+1} b\mathbf{P}_{1,x}^{m_1+i-1} \mathbf{P}_{2,x}^{m_2+1} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+i-1} \mathbf{P}_{2,y}^{m_2+1} (\mathbf{P}_{1,y} - \mathbf{P}_{2,y}) \\
& \geq 0,
\end{aligned}$$

where the first inequality is true because $\mathbf{P}_{2,x} < \mathbf{P}_{2,y}$, $\mathbf{P}_{1,x} - \mathbf{P}_{2,x} > 0$ and $\mathbf{P}_{1,y} - \mathbf{P}_{2,y} < 0$, therefore for every $1 \leq i \leq H-t+1$

$$b\mathbf{P}_{1,x}^{m_1+i-1} \mathbf{P}_{2,x}^{m_2} (\mathbf{P}_{1,x} - \mathbf{P}_{2,x})(1 - \mathbf{P}_{2,x}) > 0 > (1-b)\mathbf{P}_{1,y}^{m_1+i-1} \mathbf{P}_{2,y}^{m_2} (\mathbf{P}_{2,y} - \mathbf{P}_{1,y})(\mathbf{P}_{2,y} - 1)$$

along with (3.29). The second inequality follows from (3.28). Thus, we have success-

fully find another policy $\pi_{1:H}^{\text{new}}$ that differs from π and achieves a higher value, which is a contradiction.

next, we consider the case where π has switched from topic 1 to topic 2 at time t, i.e.,

$$\pi_{1:H} = (\pi_{1:t-2}, \underbrace{1}_{\pi_{t-1}}, \underbrace{2}_{\pi_t}, \underbrace{2, \ldots, 2}_{\pi_{t+1:H-1}}, \underbrace{2}_{\pi_H}).$$

Consider another policy $\tilde{\pi}$ (that behaves the same as π except at time step t) defined as

$$\tilde{\pi}_{1:H} = (\pi_{1:t-2}, \underbrace{1}_{\pi_{t-1}}, \underbrace{1}_{\pi_t}, \underbrace{2, \ldots, 2}_{\pi_{t+1:H-1}}, \underbrace{2}_{\pi_H}).$$

Since π is optimal, we have the difference between the value of π and $\tilde{\pi}$ to be non-negative, i.e.,

$$\mathbb{E}[V_H^{\pi}] - \mathbb{E}[V_H^{\tilde{\pi}}] = \sum_{i=1}^{H-t+1} b\mathbf{P}_{1,x}^{m_1+1}\mathbf{P}_{2,x}^{m_2+i-1}(\mathbf{P}_{2,x} - \mathbf{P}_{1,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+1}\mathbf{P}_{2,y}^{m_2+i-1}(\mathbf{P}_{2,y} - \mathbf{P}_{1,y}) \geq 0, \tag{3.30}$$

where the difference follows from the discrepancy between the two policies from time step t to H.

Consider another policy π' (that behaves the same as π except at time step H) defined as

$$\pi'_{1:H} = (\pi_{1:t-2}, \underbrace{1}_{\pi_{t-1}}, \underbrace{2}_{\pi_t}, \underbrace{2, \ldots, 2}_{\pi_{t+1:H-1}}, \underbrace{1}_{\pi_H}).$$

Since π is optimal, we have the difference between the value of π and π' to be

non-negative, i.e.,

$$\mathbb{E}[V_H^{\pi}] > \mathbb{E}[V_H^{\pi'}] \Rightarrow (1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+H-t}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) \geq b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+H-t}(\mathbf{P}_{1,x}-\mathbf{P}_{2,x}),$$

where the difference is induced by the discrepancy of the two policies from time step H. Multiplying both sides by $\mathbf{P}_{2,x} > 0$,

$$\mathbf{P}_{2,x}(1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+H-t}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) \geq b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+H-t+1}(\mathbf{P}_{1,x}-\mathbf{P}_{2,x}).$$

Using $\mathbf{P}_{2,x}(1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+H-t}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) > 0$ and $\frac{\mathbf{P}_{2,y}}{\mathbf{P}_{2,x}} \geq 1$, we get

$$(1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+H-t+1}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) \geq b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+H-t+1}(\mathbf{P}_{1,x}-\mathbf{P}_{2,x});$$

hence,

$$b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+H-t+1}(\mathbf{P}_{2,x}-\mathbf{P}_{1,x}) + (1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+H-t+1}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y}) \geq 0. \quad (3.31)$$

Again, we will construct a new policy π_{new} that outperforms π. We let π^{new} to be the policy defined as below

$$\pi_{1:H}^{\text{new}} = (\pi_{1:t-2}, \underbrace{2}_{\pi_{t-1}}, \underbrace{2}_{\pi_t}, \underbrace{2,\ldots,2}_{\pi_{t+1:H-1}}, \underbrace{1}_{\pi_H}).$$

Now, the value difference between π^{new} and π (caused by the discrepancy of the two policies from time $t-1$ to H) is

$$\begin{aligned}\mathbb{E}[V_H^{\pi^{\text{new}}}] - \mathbb{E}[V_H^{\pi}] = &\sum_{i=1}^{H-t+1}\Big(b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+i-1}(\mathbf{P}_{2,x}-\mathbf{P}_{1,x}) + (1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+i-1}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y})\Big) \\ &+ b\mathbf{P}_{1,x}^{m_1}\mathbf{P}_{2,x}^{m_2+H-t+1}(\mathbf{P}_{2,x}-\mathbf{P}_{1,x}) + (1-b)\mathbf{P}_{1,y}^{m_1}\mathbf{P}_{2,y}^{m_2+i-1}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y})\end{aligned}$$

$$> \sum_{i=1}^{H-t+1} b\mathbf{P}_{1,x}^{m_1+1}\mathbf{P}_{2,x}^{m_2+i-1}(\mathbf{P}_{2,x}-\mathbf{P}_{1,x}) + (1-b)\mathbf{P}_{1,y}^{m_1+1}\mathbf{P}_{2,y}^{m_2+i-1}(\mathbf{P}_{2,y}-\mathbf{P}_{1,y})$$
$$\geq 0,$$

where the first inequality is true because $\mathbf{P}_{1,y} < \mathbf{P}_{1,x}$, $\mathbf{P}_{2,x}-\mathbf{P}_{1,x} < 0$ and $\mathbf{P}_{2,y}-\mathbf{P}_{1,y} > 0$ and (3.31), and the second from (3.30). Similarly, we have successfully find another policy $\pi_{1:H}^{\text{new}}$ that differs from π and achieves a higher value, which is a contradiction.

We have covered all cases, so the inductive argument holds. This concludes the proof of the theorem. ∎

UCB-based regret bound

Lemma 3.48. *Let $\tilde{\tau} = \frac{8e}{\ln(\frac{1}{1-\epsilon})}$ and $\eta = 1$. For every threshold policy $\pi \in \Pi_H$, the centred random variable $V^\pi - \mathbb{E}[V^\pi]$ is (τ^2, b)-sub-exponential with (τ^2, b) satisfying $\tilde{\tau} \geq \tau$ and $\eta \geq b^2/\tau^2$.*

Proof. Let γ be such that

$$|\gamma| < -\frac{\ln(1-\epsilon)}{8e} \leq \min_{a\in\{1,2\},i\in\{x,y\}}\{-\frac{\ln(1-\mathbf{\Lambda}_{a,i}(1-\mathbf{P}_{a,i}))}{8e}\} = \min_{a\in\{1,2\},i\in\{x,y\}}\{-\frac{\ln(\mathbf{P}_{a,i})}{8e}\}.$$

Next, we have that

$$\mathbb{E}[\exp(\gamma(V^\pi - \mathbb{E}[V^\pi]))] \leq \sum_{a\in\{1,2\}} \mathbb{E}[\exp(\gamma(V^{\pi_a} - \mathbb{E}[V^{\pi_a}]))]\Big|type(t) \in \operatorname*{argmax}_{i\in[1,2]} \mathbf{P}_{a,i}]\cdot\Pr[type(t) \in \operatorname*{argmax}_{i\in[1,2]} \mathbf{P}$$

$$\max_{a\in\{1,2\}}\{\mathbb{E}[\exp(\gamma(\bar{V}^{\pi^a} - \mathbb{E}[\bar{V}^{\pi^a}]))]\}$$

Where $\bar{V}^{\pi_a}$ is the return for the instance $\langle[1],[2],\mathbf{q},\bar{\mathbf{P}},\bar{\mathbf{\Lambda}}\rangle$ such that for every $a \in \{1,2\}$: $\bar{\mathbf{P}}_{a,1} = \max_{i\in\{x,y\}} \mathbf{P}_{a,i}$ and $\mathbf{\Lambda}_{a,1} = 1$.

Finally, from Lemma 3.38 we get

$$\max_{a\in\{1,2\}}\{\mathbb{E}[\exp(\gamma(\bar{V}^{\pi^a}-\mathbb{E}[\bar{V}^{\pi^a}]))]\}\leq\max_{a\in\{1,2\}}\exp((-\frac{8e}{\ln(\bar{\mathbf{P}}_{a,1})})^2\frac{\gamma^2}{2})=\max_{a\in\{1,2\},i\in\{x,y\}}\exp((-\frac{8e}{\ln(\mathbf{P}_{a,i})})^2\frac{\gamma^2}{2}).$$

Choosing

$$\tau=b=\max_{a\in\{1,2\},i\in\{x,y\}}-\frac{8e}{\ln(\mathbf{P}_{a,i})}$$

completes the proof as

$$\max_{a\in\{1,2\},i\in\{x,y\}}-\frac{8e}{\ln(\mathbf{P}_{a,i})}\leq-\frac{8e}{\ln(1-\epsilon)}=\tilde{\tau}\quad\text{and}\quad\frac{\tau^2}{b^2}=1=\eta.$$

■

Lemma 3.47. *For every $H\in\mathbb{N}$, it holds that*

$$\mathbb{E}\left[V^{\pi^*}-\max_{\pi\in\Pi_H}V^{\pi}\right]\leq\frac{1}{2^{O(H)}}.$$

Proof. Recall that $V^{\pi}=\sum_{j=1}^{N^{\pi}}r_j(\pi_j)$, where we drop the dependence on the user index for readability. Formulating differently, for any $H\in\mathbb{N}$ it holds that

$$V^{\pi}=\sum_{j=1}^{H}\mathbb{I}_{j\leq N^{\pi}}\cdot r_j(\pi_j)+\sum_{j=H+1}^{\infty}\mathbb{I}_{j\leq N^{\pi}}\cdot r_j(\pi_j).$$

Using the same representation for $V^{\pi'}$ and taking expectation, we get that

$$\begin{aligned}\mathbb{E}\left[V^{\pi}-V^{\pi'}\right]&\leq\mathbb{E}\left[\sum_{j=1}^{H}\mathbb{I}_{j\leq N^{\pi}}\cdot r_j(\pi_j)-\sum_{j=1}^{H}\mathbb{I}_{j\leq N^{\pi'}}\cdot r_j(\pi'_j)\right]+\mathbb{E}\left[\sum_{j=H+1}^{\infty}\mathbb{I}_{j\leq N^{\pi}}\cdot r_j(\pi_j)\right]\\&\leq 0+\mathbb{E}\left[\sum_{j=H+1}^{\infty}\mathbb{I}_{j\leq N^{\pi}}\cdot r_j(\pi_j)\right]=\sum_{j=H+1}^{\infty}\Pr\left(j\leq N^{\pi}\right)r_j(\pi_j)\\&\leq\sum_{j=H+1}^{\infty}(1-\epsilon)^j(1-\epsilon)=(1-\epsilon)^{H+2}\sum_{j=0}^{\infty}(1-\epsilon)^j\end{aligned}$$

$$\leq (1-\epsilon)^H \frac{1}{\epsilon} \leq \frac{e^{-\epsilon H}}{\epsilon} = \frac{1}{2^{O(H)}}.$$

■

3.3 The SafeZone Problem

3.3.1 Introduction

Most research in reinforcement learning (RL) deals with the problem of learning an optimal policy for some Markov decision process (MDP). One notable exception for that is Safe RL, that focuses on finding the best policy that meets safety requirements. Typically, these problems are handled by adjusting the objective to include safety requirements and then optimizing over it, or incorporating additional safety constraints to the exploration stage. Anomaly Detection is the problem of identifying patterns in data that do not correspond to what is expected, i.e., anomalies. Anomaly Detection addresses a variety of applications: cyber-security, fraud detection, failure detection, etc. (see [32] for survey).

In this chapter, we introduce the SafeZone problem, a general approach for safe RL and anomaly detection that concentrates on a given policy rather than finding a policy that follows some predefined safety specifications and emphasizes entire trajectories in order to detect anomalies.

Consider a policy for a finite horizon MDP. The policy induces a Markov Chain (MC) on the MDP. Given a subset of states, we define the *escape probability* to be the probability that a random trajectory has at least one state outside this subset (hence the trajectory *escapes* it). A SafeZone is a subset of states whose quality is measured by its' size and escape probability (ideally, both are small). If a SafeZone has low escape probability, we consider it *safe*.

Trivial SafeZone solutions are the entire set of states (which has minimal escape probability of 0 on the account of maximal size), and the empty set (which has minimal size but has maximal escape probability of 1). We are interested to find SafeZone with a good tradeoff: namely a relatively small set size with small escape probability. More precisely, given a bound over the escape probability, $\rho > 0$, the goal of the learner is, using trajectory sampling, to find the smallest SafeZone with escape probability at most ρ. We address unknown environment, by which we mean no prior knowledge on the transition function or the policy used. The learner can only access random trajectories generated by the induced MC. For many applications, if there exists such a small SafeZone it is useful to find it.

Consider for example automatic robotic arm that assembles products. If something unusual happened during the assembly of a product, it might result in a malfunctioning product. In that case, the operator should be notified (anomaly detection). On the other hand, we would not like to call the operator too often. If we find a SafeZone, we can make sure that we notify the operator only in the rare events the production process (trajectory) escapes it. Furthermore, if the SafeZone is small, the manufacturer can potentially test the SafeZone states and verify their compliance, ensuring that the majority of products are well constructed for a significantly lower testing budget.

Another useful application is transportation design. For example, given data regarding bicycle commutes (not necessarily done on bicycle lanes) in a populated areas, pave bike lanes in the SafeZone, namely in a way that would accommodate popular commutes, from starting point to destination. Making cycling safer and more accessible would also promote it as a viable transportation option, which in turn benefits the environment [127].

We remark that finding a SafeZone alone does not suffice for safety; Rather, a

nearly optimal SafeZone is a behavioral description that can be used for safety applications, such as safer cycling. As another example, efficient testing (of states within the SafeZone) that "captures" most of the products' assembly process would improve safety.

Other motivations include imitation learning with compact policy representation. Namely, design a smaller state policy that preforms well for most cases but might be undefined on some states. In this case, trajectories that reach undefined states have zero reward, and such trajectories are captured by the escape probability. One natural application for that is creating a 'lite' version for a given software such as Microsoft's Windows Lite.

Our work can also be viewed through the lens of explainable RL, where the goal is to explain a specific policy. SafeZone is a new post-hoc explanation of the summarization type [3]. Going back to the bicycle example, a municipality could provide a convincing explanation to its community for the chosen design.

Our results include approximation algorithms for the SafeZone problem, which we show is NP-hard, even when the model is given and the horizon is small ($H = 2$). We are interested in a good tradeoff between the escape probability of the SafeZone and its size. Our algorithms are evaluated based on two criteria: their approximation factors (w.r.t. the escape probability bound and the optimal set size for this bound), and their trajectory sample complexity bounds (e.g., [51]).

Our results are the following:

1. Introducing the SafeZone problem (section 3.3.2), and some of its applications.
2. We explore naive approaches, namely greedy algorithms that select SafeZones based on state distributions and trajectory sampling. In addition, we show cases in which their solutions are far from optimal, either in terms of high escape

probability or significantly larger set size (see section 3.3.3).

3. We design FINDING SAFEZONE, an efficient approximation algorithm with provable guarantees. The algorithm returns a SAFEZONE which is slightly more than twice in terms of both the size and the escape probability compared to the optimal (see section 3.3.4).
4. We prove that finding a SAFEZONE is NP-hard, even for horizon $H = 2$ and known environment setting (i.e., when the induced Markov chain is given) in section 3.3.5.
5. We conclude the section with an empirical demonstration in section 3.3.6.

Trajectory escaping. The SAFEZONE problem deals escaping trajectories. In particular, given a SAFEZONE, a trajectory escapes it, no matter if only one of its states is outside the SAFEZONE or all of them. A related, yet very different problem, is that of minimizing a subset size, such that the expected number of states outside the set is minimized. This related problem, while significantly easier (as it is solved by returning the most visited states), does not apply to the applications we described earlier. In Section 3.3.3, we show that the solution for the SAFEZONE does not necessarily overlaps with the most visited states. Furthermore, simply returning states which appeared in trajectory samples could result in a set size far from optimal.

Related Work

MDPs have been studied extensively in the context of decision making in particular by the Reinforcement Learning (RL) community (see [110] for a broad background on MDPs, and [124] for background on reinforcement learning).

Safe RL. A related line of research is safe RL, where the learner's goal is to find the best policy that satisfies safety guarantees. The two main methodologies to handle

such problems are: (1) altering the objective to include the safety requirement and optimizing over it, and (2) adding safety constraints to the exploration part. See [106, 49, 131, 67, 65] for recent works and [58, 5] for surveys. In our work, the goal is not to find the optimal policy, but instead given a policy, finding its SafeZone. Moreover, the SafeZone problem is not characterized by specific requirements, and beyond the MDP, the solution could very much depend on the given policy.

Imitation Learning. In imitation learning, the learner observes a policy behaviour and wants to imitate it (see [70] for survey). Similar to imitation learning, we are given access to samples of a given policy. In contrast, rather than imitating the policy we find the policy's SafeZone, which is an important property of the policy.

Approximate MDP equivalence. Another related research line is that of finding an (almost) equivalent minimal model for a given MDP, where the goal is that the optimal policy on the (almost) equivalent model induces an (approximately) optimal policy in the original MDP, e.g., [60, 53]. This line of works and ours differ in that we do not try to modify the MDP (e.g., cluster similar states), but rather to find a SafeZone, a property which is defined for the existing MDP and a specific policy.

Explainability. In explainability, the goal is to provide a post-hoc explanation to a specific given model [101], e.g., using decision trees [21, 102], influential examples [85], or a local approximation explanations [91]. We focus on explainability for reinforcement learning, and specifically we suggest a new summarization explanation through our SafeZone, [4].

MC with traps. A decision problem that might seem related to ours is that of MC with traps ([46]): Given an input of a MC (with possibly infinite state space), a starting state, and states trapping (absorbing) probabilities, the goal is to decide

whether or not a (possibly infinite) random walk would reach an absorbing state with probability 1, or not. In section 3.3.9, we explain why this problem is inherently different than SafeZone.

3.3.2 SafeZone: Problem Formulation

We model the problem using a Markov model with finite horizon $H > 1$. Formally, there is a Markov chain (MC) $\langle \mathcal{S}, P, s_0 \rangle$ where $\mathcal{S}$ is the set of states, $s_0 \in \mathcal{S}$ is the initial state and $P : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is the transition function that maps a pair of states into probability by $P(s, s') = \Pr[s_{t+1} = s' | s_t = s]$. We assume the transition function P is induced by a policy $\pi : \mathcal{S} \to \text{Simplex}^{\mathcal{A}}$ on a MDP $\langle \mathcal{S}, s_0, P', \mathcal{A} \rangle$ with transition function $P' : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ such that $P(s, s') = \sum_{a \in \mathcal{A}} P'(s, a, s') \cdot \pi(a|s)$ for all $s, s' \in \mathcal{S}$ (though any MC can be generated this way, thus our theoretical guarantees apply for general MCs).

A *trajectory* $\tau = (s_0, \ldots, s_H)$ starts in the initial state s_0 and followed by a sequence of H states generated by P, i.e., $\Pr[s_{i+1} = s' | s_i = s] = P(s, s')$ for all $i \in [H]$, where $[H] := \{1, \ldots, H\}$. We abuse the notation and regard a trajectory τ both as a sequence and a set.

Given a subset of states $F \subseteq \mathcal{S}$, a trajectory τ *escapes* F if it contains at least one state $s \in \tau$ such that $s \notin F$, i.e., $\tau \not\subseteq F$. We refer to the probability that a random trajectory escapes F as *escape probability* and denote it by $\Delta(F) = \Pr_\tau[\tau \not\subseteq F]$. We call F a $\rho-$*safe* (w.r.t. the model $\langle \mathcal{S}, s_0, P \rangle$) if its escape probability, $\Delta(F)$, is at most ρ. Formally,

Definition 3.58. *A set $F \subseteq \mathcal{S}$ is $\rho-$safe if*

$$\Delta(F) := \Pr_\tau[\tau \not\subseteq F] \leq \rho,$$

where τ is a random trajectory.

A set $F \subseteq \mathcal{S}$ is called $(\rho, k)-$*SafeZone* if F is $\rho-$safe and $|F| \leq k$. Given a safety parameter $\rho \in (0, 1)$, we denote the smallest size $\rho-$safe set by $k^*(\rho)$:

$$k^*(\rho) = \min_{F \subseteq \mathcal{S} \text{ is } \rho-\text{safe}} |F|.$$

Whenever the discussed parameter ρ is clear from the context we use k^* instead of $k^*(\rho)$. We remark that there might be multiple different $(\rho, k)-$SafeZone sets.

The learner knows the set of states, $\mathcal{S}$, the initial state, s_0, and the horizon H but has no knowledge regarding the transition function P or the minimal size of the $\rho-$safe set, k^*. Instead, the learner receives information about the model from sampling trajectories from the distribution induced by π.

Given $\rho > 0$, the ultimate goal of the learner would have been to find a $(\rho, k^*(\rho))-$SafeZone. However, as we show in section 3.3.5, finding a $(\rho, k^*(\rho))-$SafeZone is NP-hard, even when the transition function P is known. This is why we loosen the objective to find a bi-criteria approximation $(\rho', k')-$SafeZone . (Bi-criteria approximations are widely studied in approximation and online algorithms [125, 129].) In our setting, given ρ the objective is to find a set F which is $(\rho', k')-$SafeZone with minimal size $k' \geq k^*$ and minimal escape probability $\rho' \geq \rho$. In addition, we are interested in minimizing the sample complexity.

Notice that the learner can efficiently verify, with high probability, whether a set F is approximately $\rho-$safe or not. The following proposition formalize this and follows directly from lemma 3.75.

Proposition 3.59. *There exists an efficient algorithm such that for every set $F \subseteq \mathcal{S}$ and parameters $\rho, \lambda > 0$, the algorithm samples $O(\frac{1}{\epsilon^2} \ln \frac{1}{\lambda})$ random trajectories and*

Table 3.3: Upper bounds for safety and size. * Only for layered MDPs.

Algorithm	**Safe**	**Set Size**	**Sample Complexity**
Greedy by Threshold	2ρ	k^*H/ρ	–
Simulation	2ρ	$O(k^*H \ln k^*)$	$poly(k^*, \frac{1}{\rho})$
Greedy at Each Step*	ρH	k^*	–
Finding SafeZone	$2\rho + 2\epsilon$	$(2+\delta)k^*$	$poly(k^*, H, \frac{1}{\epsilon}, \frac{1}{\delta})$

returns $\widehat{\Delta}(F)$*, such that with probability at most* λ *we have* $|\Delta(F) - \widehat{\Delta}(F)| \geq \epsilon$.

Summary of Contributions. We summarize the results of all the algorithms that appear in the section in Table 3.3. The bounds of Greedy by Threshold and Greedy At Each Step requires the Markov Chain model as input, and a pre-processing step that takes $O(|S|^2H)$ time. Additionally, the bounds for first three algorithms (the naive approaches) requires an additional knowledge of $k^*(\rho)$. Beyond the upper bounds, we provide instances that show that the upper bounds are tight up to a constant for each of the first three algorithms (the naive approaches). The following theorem is an informal statement of our main theorem, theorem 3.66.

Theorem 3.60. *For every* $\rho, \epsilon, \delta > 0$*, with probability* ≥ 0.99 *there exists an algorithm that returns a set which is* $(2\rho + 2\epsilon, (2+\delta)k^*) -$ *SafeZone.*

The running time of the algorithm is also bounded by $poly(k^*, H, \frac{1}{\delta}, \frac{1}{\epsilon})$. We empirically evaluate the suggested algorithms on a grid-world instance (where the goal is to reach an absorbing state), showing that Finding SafeZone outperforms the naive approaches. Moreover, we show that different policies have qualitatively different SafeZones. Finally, an informal statement of theorem 3.68.

Theorem 3.61. *SafeZone is NP-hard.*

3.3.3 Gentle Start

This section explains and analyzes various naive algorithms to the SafeZone problem. We show that even if the transition function is known in advance, these naive algorithms result in outputs that are far from optimal. To describe the algorithms, we define for each state s the probability to appear in a random trajectory and denote it by $p(s) = \Pr_\tau[s \in \tau] \in [0,1]$. Note that $\sum_{s\in\mathcal{S}} p(s)$ is a number between 1 and H (e.g., $p(s_0) = 1$), and can be estimated efficiently using dynamic programming if the environment and policy are known and sampling otherwise. To be precise, some of the algorithms assume the probabilities $\{p(s)\}_{s\in\mathcal{S}}$ are received as input.

Greedy by Threshold Algorithm. The algorithm gets, in addition to ρ, the distribution p and a parameter $\beta > 0$ as input. It returns a set F that contains all states s with probability at least β, i.e., $p(s) \geq \beta$. We formalize this idea as algorithm 14 in Section 3.3.10. For $\beta = \frac{\rho}{k^*}$, the output of the algorithm is $\left(2\rho, \frac{k^*H}{\rho}\right) - \text{SafeZone}$. More generally, we prove the following lemma.

Lemma 3.62. *For any $\rho, \beta \in (0,1)$, the Greedy by Threshold Algorithm returns a set that is $(\rho + k^*\beta, \frac{H}{\beta}) - $ SafeZone. In particular, for $\beta = \frac{\rho}{k^*}$, this set is $\left(2\rho, \frac{k^*H}{\rho}\right) - $ SafeZone.*

While it is clear why there are instances for which the safety is tight, Lemma 3.71 in Section 3.3.10 shows that the set size is tight as well.

Simulation Algorithm. The algorithm samples $O(\frac{\ln k^*}{\beta})$ random trajectories and returns a set F with all the states in theses trajectories. It is formalized in Section 3.3.10 as algorithm 15.

Lemma 3.63. *Fix $\rho, \beta \in (0,1)$. With probability at least 0.99, Simulation Algorithm returns a set that is $\left(\rho + k^*\beta, O(k^* + \frac{\rho H \ln k^*}{\beta})\right) -$ SafeZone. In particular, for $\beta = \frac{\rho}{k^*}$,*

this set is $(2\rho, O(k^* H \ln k^*)) - \textit{SafeZone}$.

While this algorithm achieves a low escape probability, only 2ρ, in Lemma 3.72 in Section 3.3.10 we prove that the size of F is tight up to a constant, i.e., an MDP instance where $|F| = \Omega(k^* H \ln k^*)$.

So far, the presented algorithms were approximately safe (i.e., low escape probability), but might return large subsets. Without further assumptions, the following algorithm provides a $(\rho H, Hk^*)-$SafeZone. However, when considering MDPs with a special structure it provides an optimal sized SafeZone , at the price of large escape probability.

Greedy at Each Step Algorithm. For the analysis of the next algorithm we assume the MDP is *layered*, i.e., there are no states that appear in more than a single time step and denote $\mathcal{S} = \bigcup_{i=1}^{H} \mathcal{S}_i$. I.e., the transitions $P(s, s')$ are nonzero only for $s' \in \mathcal{S}_{i+1}$ and $s \in \mathcal{S}_i$. The Greedy at Each Step Algorithm takes at each time step i the minimal number of states such that the sum of their probabilities is at least $1 - \rho$. It is formalized in Section 3.3.10 as algorithm 16.

Lemma 3.64. *For any $\rho \in (0, 1)$, if the MDP is layered, Greedy at Each Step Algorithm returns a set that is $(\rho H, k^*) - \textit{SafeZone}$.*

In Lemma 3.73 we have a lower bound on the escape probability, which asymptotically matches.

Weaknesses of the naive algorithms. We showed algorithms that identify SafeZone with either escape probability much greater than ρ or with size much greater than k^*. This holds even when providing extra information (such as the transition function and/or the optimal size of the $\rho-$safe set, i.e., k^*). Moreover, we showed tight lower bounds for these algorithms.

3.3.4 Algorithm for Detecting SafeZones

In this section we suggest a new algorithm that builds upon and improves the added trajectory selection of the SIMULATION Algorithm. One reason for why SIMULATION returns a large set is that it treats every sampled trajectory identically, regardless of how many states are being added.

More precisely, fix any $(\rho, k^*)-$SAFEZONE set, F^*, and consider a trajectory τ that escapes it, i.e., $\tau \not\subseteq F^*$. If τ was sampled, its states are added to the constructed set F, which might increase the size of F by up to H states that are not in F^*, without significantly improving the safety.

In contrast, when selecting which trajectory to add to F, we would consider the number of states it adds to the current set. For the sake of readability, we refer to any state which is not in the current set F as *new*, and denote by $new_F(\tau)$ the number of new states in τ w.r.t. F, i.e.,

$$new_F(\tau) := |\tau \setminus F|.$$

Note that for every $F \subseteq \mathcal{S}$, we have that $\Pr_\tau[new_F(\tau) \neq 0] = \Delta(F)$.

The new algorithm does not sample each trajectory uniformly at random, but sample from a new distribution, which will be denoted by Q_F. While favoring trajectories with higher probabilities, which we already get by the sampling process, another key idea would guide this new distribution: To prefer trajectories that *gradually* increase the size of F. To implement this idea, we will ensure that the probability of adding a trajectory τ to F should be *inversely proportional* to $new_F(\tau)$.

Formally, the support of Q_F is the trajectories with new states, i.e., $X = \{\tau | new_F(\tau) \neq 0\}$. For every $\tau \in X$, $Q_F(\tau) \propto \frac{\Pr[\tau]}{new_F(\tau)}$, where $\Pr[\tau]$ is the probability of trajectory τ

under the Markov Chain with dynamics P. Note that the new distribution depends on the current set F, and changes as we modify it. Intuitively, adding trajectories to F according to Q_F instead of adding trajectories sampled directly from the dynamics (as we do in SIMULATION) would increase the expected ratio between the added safety and the number of new states we add to F, thus improving the set size guarantee of the output set. We elaborate on this in section 3.3.4.

Our main algorithm is FINDING SAFEZONE, Algorithm 12. The algorithm receives, in addition to the safety parameter ρ, parameters $\epsilon, \lambda \in (0,1)$, and maintains a set F that is initiated to $\{s_0\}$. On a high level, to implement the idea of adding trajectories to F according to Q_F, we use *rejection sampling*. Namely, in each iteration of the while–loop we first sample a trajectory τ and if $new_F(\tau) \neq 0$, we *accept* it with probability $1/new_F(\tau)$. If the trajectory is accepted, it is added to F. More precisely, if $new_F(\tau) \neq 0$, we sample a Bernoulli random variable, $accept \sim Br(1/new_F(\tau))$. If $accept = 1$, we add τ to F. This process of adding trajectories to F generates the desired distribution, Q_F.

Whenever a trajectory is added to F, we estimate the escape probability $\Delta(F)$ (w.r.t. the updated set, F). The algorithm stops adding states to F and returns it as output when it becomes "safe enough". To be precise, let $\widehat{\Delta}(F)$ denote the result of the escape probability estimation (by sampling trajectories as suggested in Proposition 3.59). If $\widehat{\Delta}(F) \leq 2\rho + \epsilon$, it means that F is $(2\rho + 2\epsilon)-$safe with probability $\geq 1 - \lambda_j > 1 - \lambda$, in which case the algorithm terminates and returns F as output. To implement the estimation $\widehat{\Delta}(F)$, the algorithm calls *EstSafety* Subroutine. The subroutine samples $N_j = \Theta(\frac{1}{\epsilon^2} \ln \frac{2}{\lambda_j})$ trajectories, and returns the fraction of trajectories that escaped F.

For cases in which the transition function P is known to the learner, we provide an alternative implementation for *EstSafety* which computes the exact probability $\Delta(F)$

(see section 3.3.8).

Algorithm Analysis

We define the event

$$\mathcal{E} = \{\forall i \; |\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})| \leq \epsilon\},$$

which states that all our *EstSafety* Subroutine estimations are accurate. We show that $\mathcal{E}$ holds with high probability using Hoeffding's inequality. In most of the analysis we condition on $\mathcal{E}$ to hold.

The following theorem is the central component in the proof of the main theorem that follows it.

Theorem 3.65. *Given $\rho, \epsilon, \lambda \in (0,1)$, Finding SafeZone Algorithm returns a subset $F \subseteq \mathcal{S}$ such that:*

1. *The escape probability is bounded from above by $\Delta(F) \leq 2\rho + 2\epsilon$, with probability $1 - \lambda$.*
2. *The expected size of F given $\mathcal{E}$ is bounded by $\mathbb{E}[|F| \mid \mathcal{E}] \leq 2k^*$.*
3. *The sample complexity of the algorithm is bounded by $O\left(\frac{k^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda} + \frac{Hk^*}{\rho\lambda}\right)$, and the running time is bounded by $O\left(\frac{Hk^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda} + \frac{H^2k^*}{\rho\lambda}\right)$, with probability $1 - \lambda$.*

To obtain the main theorem, we run Finding SafeZone Algorithm several times and return the smallest output set, F, see the next section for more details.

Theorem 3.66. *(main theorem) Given $\epsilon, \rho, \delta > 0$, if we run Finding SafeZone for $\Theta(\frac{1}{\delta})$ times and return the smallest output set, $F \subseteq \mathcal{S}$, then with probability ≥ 0.99*

1. *The escape probability is bounded by $\Delta(F) \leq 2\rho + 2\epsilon$.*
2. *The size of F is bounded from above by $|F| \leq (2+\delta)k^*$.*
3. *The total sample complexity and running time are bounded by $O(\frac{k^*}{\delta^2\epsilon^2}\ln\frac{k^*}{\delta} + \frac{Hk^*}{\rho\delta^2})$,*

and $O(\frac{Hk^*}{\delta^2\epsilon^2}\ln\frac{k^*}{\delta} + \frac{H^2k^*}{\rho\delta^2})$*, respectively.*

Algorithm 12 FINDING SAFEZONE

Input: $\rho \in (0,1)$

Parameters: $\epsilon, \lambda \in (0,1)$

$F \leftarrow \{s_0\}, j \leftarrow 1, \widehat{\Delta}(F) \leftarrow 1$

while $\widehat{\Delta}(F) > 2\rho + \epsilon$ **do**

  $\tau \leftarrow$ sample a random trajectory

  Compute $new_F(\tau)$

  if $new_F(\tau) \neq 0$ **then**

    sample $accept \sim Br(1/new_F(\tau))$

    if $accept = 1$ **then**

      $F \leftarrow F \cup \tau$

      $\lambda_j \leftarrow \frac{3\lambda}{2(j\pi)^2}$, $j \leftarrow j+1$

      $\widehat{\Delta}(F) \leftarrow EstSafety(\epsilon, \lambda_j, F)$

    end if

  end if

end while

return F

Algorithm 13 *EstSafety* Subroutine

Input: subset F

Parameters: $\epsilon, \lambda_j \in (0,1)$

$\widehat{\Delta}(F) \leftarrow 0$

$\mathcal{T} \leftarrow$ sample $N_j = \frac{1}{2\epsilon^2}\ln\frac{2}{\lambda_j}$ trajectories

for $\tau \in \mathcal{T}$ **do**

  if $\tau \not\subseteq F$ **then**

    $\widehat{\Delta}(F) \leftarrow \widehat{\Delta}(F) + \frac{1}{N_j}$

  end if

end for

return $\widehat{\Delta}(F)$

Proof Technique

Escape probability set size bounds. To ease the presentation of the proof, we assume that $\widehat{\Delta}(F) = \Delta(F)$. This case is interesting by its own, since if the policy and transition function are known, we can compute $\Delta(F)$ efficiently using dynamic programming (see section 3.3.8). As a result, event $\mathcal{E}$ always holds. In addition, it is clear that the termination of the algorithm implies that $\widehat{\Delta}(F) = \Delta(F) \leq 2\rho$, thus F is $(2\rho + 2\epsilon)-$safe. The main challenge is bounding the size of F.

A few notations before we start. Let F^* denote a minimal $\rho-$safe set (of size k^*). Consider iteration i inside the while–loop. The random variable G_i is the number of states in F^* that are added to F in iteration i and B_i is the number of states added to F in iteration i that are not in F^* (G stands for *good* and B for *bad*). Notice that both G_i and B_i depend on the current set F. Notice that the size of the output set is exactly $\sum_i B_i + G_i$ and that $\sum_i G_i \leq k^*$.

The main idea of the proof technique is to show that by adding trajectories according to the new distribution Q_F, we ensure that, in expectation, there are at least as much good states that are added to F as bad states. Suppose the trajectory τ was chosen to be added to F^* by the algorithm. If $\tau \subseteq F^*$, then G_i is equal to $new_F(\tau)$ and $B_i = 0$. If $\tau \not\subseteq F^*$, then $B_i \leq new_F(\tau)$. Summarizing these observations, we have the following bounds

$$G_i \geq new_F(\tau) \cdot \mathbb{I}[\tau \subseteq F^*] \text{ and } B_i \leq new_F(\tau) \cdot \mathbb{I}[\tau \not\subseteq F^*],$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Moreover, a direct consequence of the probability in which τ is added to F is that for any set of trajectories T it holds that

$$\begin{aligned} &\mathbb{E}_{\tau \sim Q_F}[new_F(\tau) \cdot \mathbb{I}[\tau \in T]] = \sum_{\tau \in T} Q_F(\tau) new_F(\tau) \\ &= \frac{1}{Z} \sum_{\tau \in T, new_F(\tau) \neq 0} \left(\frac{\Pr[\tau]}{new_F(\tau)} \right) new_F(\tau) = \frac{1}{Z} \Pr_{\tau}[\tau \in T \wedge new_F(\tau) \neq 0], \end{aligned} \tag{3.32}$$

where Z is the normalization factor of Q_F.

To bound the size of F, we want to show that the algorithm does not add too many states outside of F^*. We therefore bound $\mathbb{E}[B_i]/\mathbb{E}[G_i]$, where the expectations are over the trajectory τ that is added to F according to Q_F. Applying Equation (3.32)

twice, once with $T = \{\tau \mid \tau \subseteq F^*\}$ and once with $T = \{\tau \mid \tau \not\subseteq F^*\}$, we bound the ratio between B_i and G_i by

$$\frac{\mathbb{E}[B_i]}{\mathbb{E}[G_i]} \leq \frac{\Pr_\tau[\tau \not\subseteq F^* \wedge new_F(\tau) \neq 0]}{\Pr_\tau[\tau \subseteq F^* \wedge new_F(\tau) \neq 0]}. \tag{3.33}$$

We know that $\Pr_\tau[\tau \not\subseteq F^*]$ is always smaller than ρ, so the numerator is $\leq \rho$. A lower bound for the denominator is

$$\Pr_\tau[new_F(\tau) \neq 0] - \Pr_\tau[\tau \not\subseteq F^*]. \tag{3.34}$$

Whenever the algorithm is inside the main loop, the safety is at least $\Pr_\tau[new_F(\tau) \neq 0] = \Delta(F) > 2\rho$. Thus (3.34) is lower bounded by ρ, and overall (3.33) is less or equal to 1, which implies that

$$\mathbb{E}[B_i] \leq \mathbb{E}[G_i]. \tag{3.35}$$

This completes the proof because we know that the algorithm does not add too many states outside of F^*. More precisely,

$$\mathbb{E}[|F|] = \mathbb{E}\left[\sum_i B_i + G_i\right] \leq \mathbb{E}\left[2\sum_i G_i\right] \leq 2k^*.$$

Sample complexity. To discuss the sample complexity, we drop the assumption that the MC is known to a learner, and uses *EstSafety* Subroutine to approximate $\Delta(F)$. The number of calls to *EstSafety* is bounded by the size of the output set, F. Hence, this part of the sample complexity is bounded by $|F| \cdot N_{|F|}$ and we show that is $O(\frac{k^*}{\epsilon^2} \log k^*)$. Another source of sampling is trajectories sampled for purposes of potentially adding them to F. Observe that at any iteration the set F has escape probability of at least 2ρ, and each trajectory that escapes F is accepted

with probability at least $1/H$. This implies a lower bound for the probability that a random trajectory is accepted is $2\rho/H$. This gives an upper bound of $\frac{2|F|\rho}{H}$ for the expected sample complexity.

Amplification. theorem 3.65 shows that if $\mathcal{E}$ holds, then the set size, $|F|$, is bounded *in expectation* by $2k^*$. As $\Pr[\mathcal{E}] \geq 1 - \lambda$ implies, from Markov's inequality, that the size $(2+\delta)k^*$ with small probability of about $\delta + \lambda = O(\delta)$. If we want to make sure that the actual size is at most $(2+\delta)k^*$ with high probability, we can repeat the process about $\Theta\left(\frac{1}{\delta}\right)$ times and take the smallest size set.

For full proofs we refer to section 3.3.11.

3.3.5 Hardness

In this section we show that SafeZone is NP-hard to solve, and this is why approximation is necessarily. Moreover, SafeZone is hard even if the MC and optimal $\rho-$safe size, k^* is known. Our starting point is the NP-hardness of regular cliques. The RegularClique(G, k_c) problem gets as an input (i) a regular graph G with n nodes where each node has degree d, and (ii) an integer k_c. It returns whether G contains a clique of size k_c. Whenever G and k_c are clear from the context we simply write RegularClique. The following fact follows, e.g., from [23].

Fact 3.67. *RegularClique is NP-hard.*

Markov chain (random walk). Fix a graph $G = (V, E)$ and a starting vertex $v_0 \in V$. The graph induces a Markov Chain (random walk) in the following way. The states of the process correspond to the vertices V in the graph G. The transition function is defined as $P(v|u) = \frac{1}{d} \cdot \mathbb{1}[(u, v) \in E]$, where d is the degree any node. The process starts from node v_0 and then proceeds according to the transition function P for H steps.

Reduction. To prove the hardness of SafeZone , we show how to solve RegularClique given a solver to SafeZone. For each vertex $v \in V$, run an algorithm for SafeZone with horizon $H = 2$, $k = k_c$, and $\rho = 1 - \left(\frac{k_c - 1}{d}\right)^2$, and v as the starting state. If there is at least one run of the algorithm that returns YES, then the final answer is YES. Otherwise, the answer is NO. Note that this reduction is efficient.

Theorem 3.68. *For every graph $G = (V, E)$ and an integer k_c there exists a clique of size k_c in $G \iff$ SafeZone$(M(G), k_c, \rho)$ answers YES.*

Given an environment, a policy and SafeZone , one could compute exactly how much safe it is (see section 3.3.8 for details), from which we deduce our next corollary.

Corollary 3.69. *SafeZone is NP-complete.*

Note that for $H = 1$, the Greedy at Each Step Algorithm is optimal.

3.3.6 Empirical Demonstration

Each of the naive approaches in Section 3.3.3, has a specific instance that the naive approach is guaranteed (w.h.p.) to return a solution which is far from optimal, as we show in section 3.3.10. The purpose of this section is to demonstrate, using a simple standard setup that, Finding SafeZone outperforms the both Greedy by Threshold and Simulation (in accordance with our theory) [9]. Additional figures and a visual comparison of two policies' different SafeZones can be found in section 3.3.7.

The MDP. We focus on a simple $N \times N$ grid problem, for some parameter N. The agent starts off at mid-left state, $(0, \lfloor \frac{N}{2} \rfloor)$ and wishes to reach the (absorbing) goal state at $(N - 1, \lfloor \frac{N}{2} \rfloor)$ with minimal number of steps. At each step it can take one

[9] As the MDP in this setup is not layered, we do not test Greedy by Each Step algorithm.

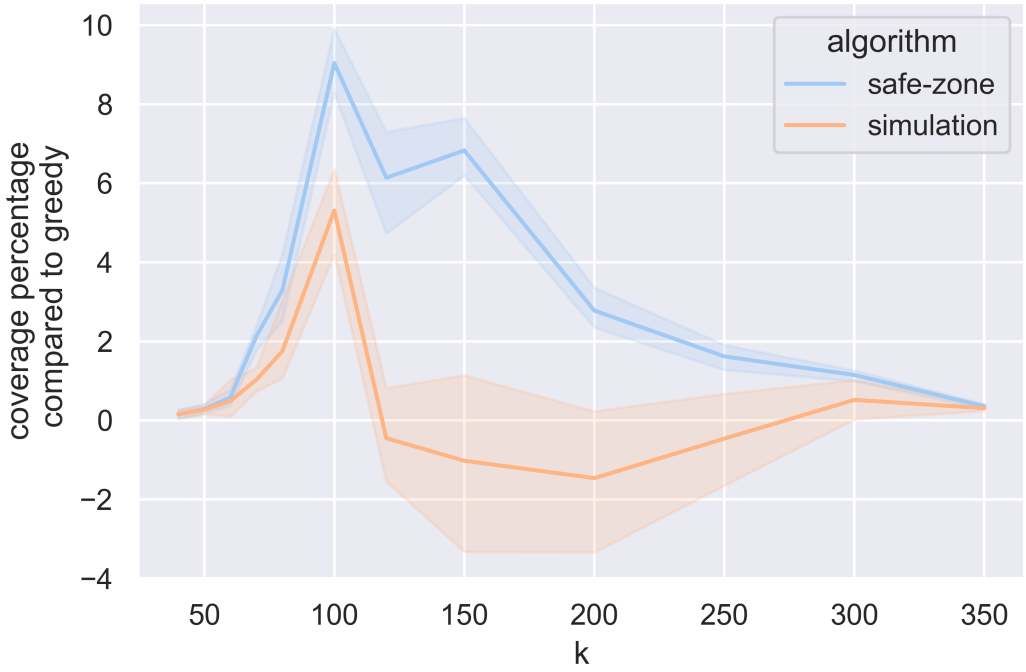


Figure 3.2: Empirical results regarding Coverage (safety) differences between the algorithms Finding SafeZones (safe-zone) and Simulation and Greedy by Threshold. The coverage of Greedy by Threshold for $k \leq 100$ is negligible (not more than 1%). For $k = 150, 200, 250, 300$, Greedy by Threshold obtains $30\%, 63\%, 83\%, 94\%$ coverage, respectively, and for $k = 350$ all algorithms obtain 100% coverage.

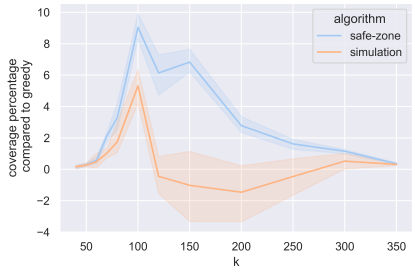
of four actions: {'up', 'down', 'right', 'left'} by 1 grid square. With probability 0.9, the intended action is performed and with probability 0.1 there is a drift down. The agent stops either way after $H = 300$ steps.

Finding SafeZone vs. naive approaches

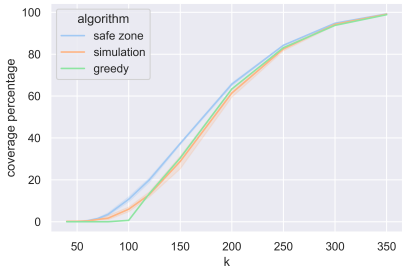
To compare the Finding SafeZone Algorithm to the naive approaches, we focus on the policy that first goes to the right and when it reaches the rightmost column, it goes up (see Figure 3.4(a) and Figure 3.3(c) in section 3.3.7 for depictions of the

number of total visits at each state using the described policy, respectively). We take $N = 30$ and 2000 episodes (i.e., the coverage (safety) of each algorithm is estimated for 2000 random trajectories). [10] fig. 3.2 depicts the trajectories coverage of each algorithm minus the coverage of the GREEDY BY THRESHOLD algorithm. A figure with the absolute values can be found in section 3.3.7 (fig. 3.3(b)). We see that the new algorithm exhibits better performance compared to its competitors. Moreover, taking less than 30% of the states ($k = 250$ out of 900 states) is enough to get a coverage of more 80% the trajectories. In section 3.3.7, we show a second policy which is slightly less optimal than this one in terms of the expected number of steps to reach to the goal state. The two policies have a very different SAFEZONES and we can clearly see that the second policy requires less states to achieve the same level of safety.

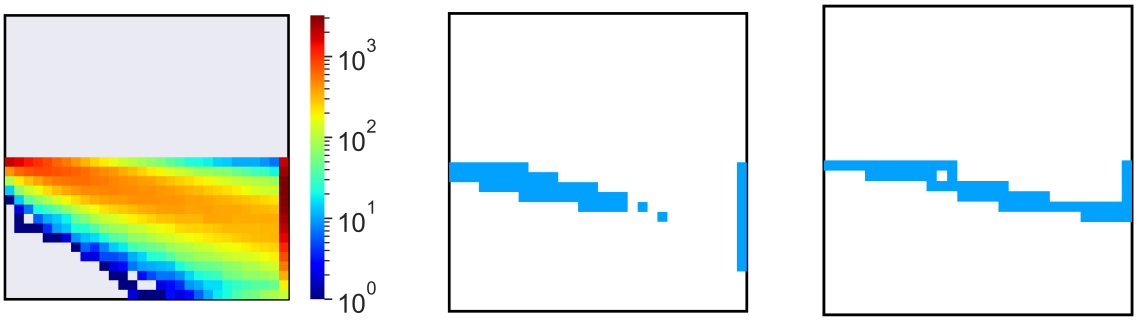
3.3.7 Extension: Additional Figures (Section 3.3.6)



(a) %Coverage: difference from GREEDY BY THRESHOLD Algorithm.



(b) %Coverage: absolute values.



(c) Total number of visits at each state from 2000 episodes. Zero visits in grey.

(d) In blue: set chosen by Greedy by Threshold Algorithm.

(e) In blue: set chosen by SafeZone Algorithm.

Figure 3.3: Empirical results regarding Coverage of the different algorithms, Finding SafeZones and state visit frequency.

Solution illustrations

Figures 3.3(d),3.3(e) show the sets found for $k = 60$ both by the *Finding SafeZone* Algorithm and Greedy by Threshold. We see that Greedy by Threshold choose an unconnected set for this small k, leading to a coverage (safety) of 0. While the new algorithm, choose a few states which consists of a several trajectories, thus leading to a coverage (safety) larger than 0.

Comparing SafeZone of two policies

In this section we empirically explore the SafeZone of two different policies within the same MDP. The first policy, described in the previous section, first goes right and then to the middle, and the second policy first goes to the middle and then goes right. See fig. 3.4. These seemingly similar policies induce very different SafeZones as can be seen in fig. 3.6 that depict the number of visits in each state. We clearly see that

[10] To illustrate the algorithm's performance, we have changed the stopping condition in their implementations from the desired safety level to desired set size, deciding randomly between different states of the trajectory in case the set side exceeds k. For Greedy By Threshold, we gradually decrease the threshold β until the set contains the desired amount of states.

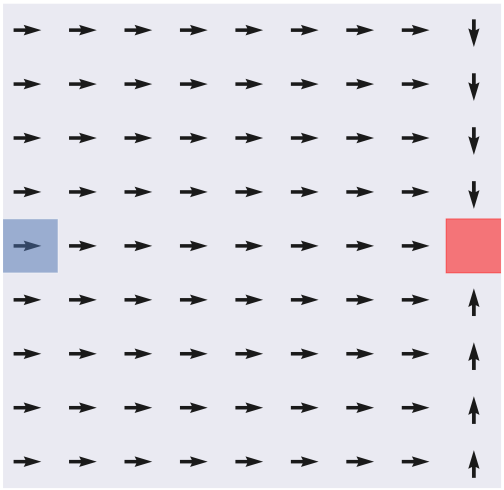
the second policy requires less states to achieve the same level of safety, even though in terms of minimizing the number of steps to get to the goal state it is outperformed by the first policy (intuitively, the second policy have more fail attempts to go up in expectation since the lowest row of the grid cannot get worst). In Figure 3.5 we see that already with 14% of the states, all three algorithms achieve trajectory coverage of more than 85%.

fig. 3.6 shows the visits of the policies described in the main section for $N = 30$. It is immediately clear that the SafeZone of the two policies are fundamentally different. As mentioned, this affects their SafeZone sizes. Namely, when trying to go right from a current state in the lowest row it is impossible to get to square which is lower than that, and the first policy takes advantage of this. In contrast, the second policy keeps trying to go up from lowest row, which implies that in expectation it goes down more times compared to the first.

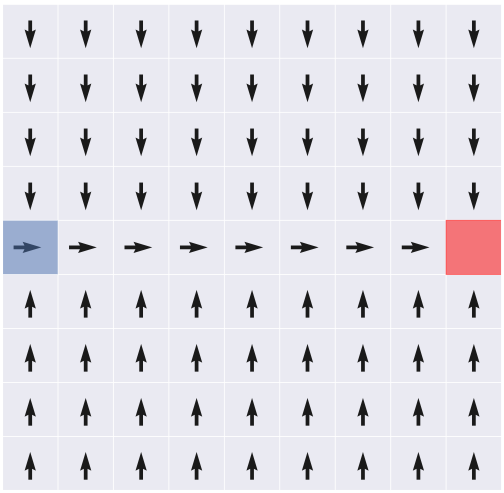
fig. 3.4 depicts the two policies discussed in the section when $N = 7$.

fig. 3.5 depicts coverage percentage for the different algorithms discussed in the section when applied to the second policy.

Similarly to Figures 3.3(d) and 3.3(e), we provide for completeness, the same figures for the policy "Go to the middle and then right". Namely, Figures 3.7(a),3.7(b) show the sets found for $k = 60$ both by the *Finding SafeZone* and Greedy by Threshold algorithms w.r.t. this policy.



(a) Go right and then to the goal state.



(b) Go to the middle and then right.

Figure 3.4: Two policies for the same MDP with $N = 7$. Starting state, s_0, in blue, goal state in red.

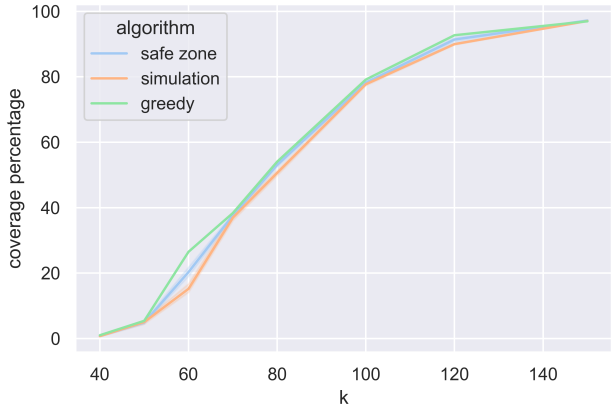
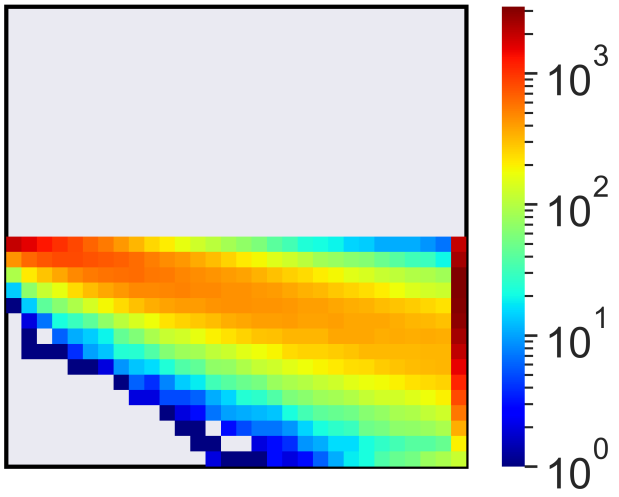
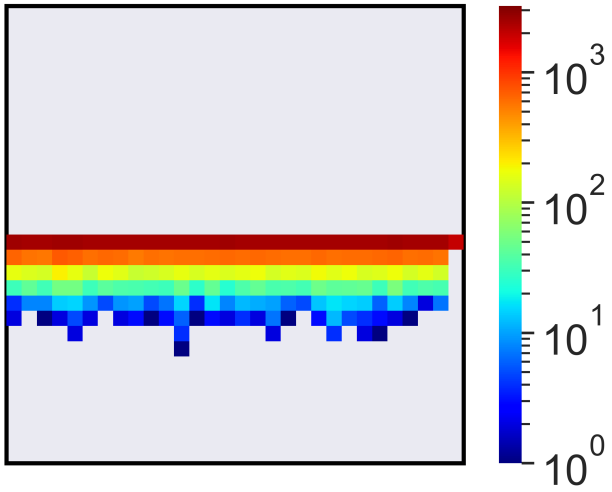


Figure 3.5: SafeZone coverage for the second policy.

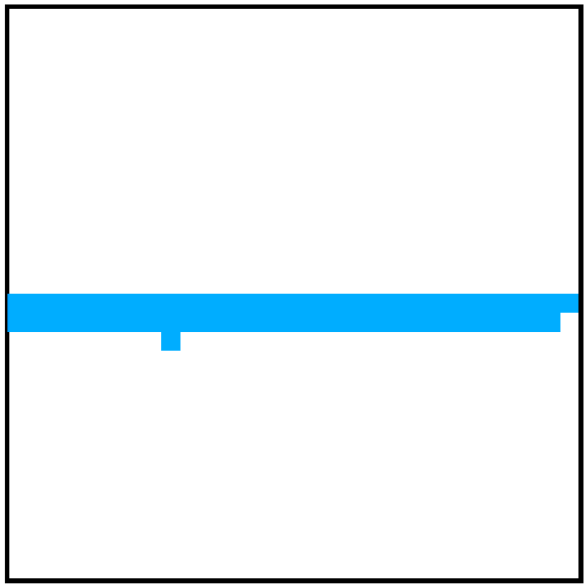


(a) Number of visits at each state for policy "Go right and then to the middle"

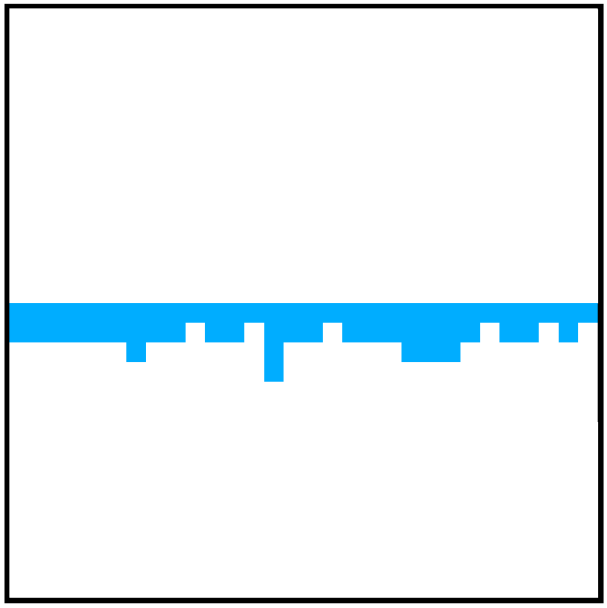


(b) Number of visits at each state for policy "Go to the middle and then right"

Figure 3.6: Total number of visits for the two policies.



(a) In blue: set chosen by Greedy at Each Step Algorithm on the policy "Go to the middle and then right"



(b) Number of visits at each state for policy "Go to the middle and then right"

Figure 3.7: Empirical results regarding Coverage of the different algorithms, Finding SafeZones and state visit frequency.

3.3.8 Extension: Exact Computation

In this section we assume that the transition function is known to the algorithm and show how to compute $\Delta(F)$.

Given a Markov Chain $\langle \mathcal{S}, P, s_0 \rangle$ and a set $F \subseteq \mathcal{S}$ we create a new Markov Chain $\langle \mathcal{S}', P', s_0 \rangle$ as follows. We add a new state $s_{sink} \notin \mathcal{S}$, and set $\mathcal{S}' = F \cup \{s_{sink}\}$. Each transition from a state $s \in F$ to a state $s' \notin F$ we modify and make the transition in P' to the sink s_{sink}. In P', when we are in s_{sink} we always stay in s_{sink}. More formally: (1) if $s, s' \in F$ then $P'(s'|s) = P(s'|s)$, (2) we set $P'(s_{sink}|s) = \sum_{s' \notin F} P(s'|s)$ and (3) $P'(s_{sink}|s_{sink}) = 1$ and $P'(s|s_{sink}) = 0$ for $s \neq s_{sink}$.

Now we claim that $\Delta(F) = \Pr_{P'}[s_H = s_{sink}]$, since any trajectory that reaches a state not in F will reach the sink in P' and stay there. We can compute $\Pr_{P'}[s_H = s_{sink}]$ using standard dynamics programming.

The running time of constructing $\langle \mathcal{S}', P', s_0 \rangle$ is $O(|\mathcal{S}|^2)$. Computing the probability of $\Pr_{P'}[s_H = s_{sink}]$ takes $O(H|\mathcal{S}|^2)$. Therefore we have established the following.

Lemma 3.70. *Given a Markov chain $\langle \mathcal{S}, P, s_0 \rangle$ and a set $F \subseteq \mathcal{S}$ we can compute $\Delta(F)$ in time $O(|\mathcal{S}|^2 H)$.*

Note that the above lemma implements an exact version of the *EstSafety* Subroutine.

3.3.9 Extension: The relation to MC with Traps

Consider a MC with countable state space and fixed absorbing probability $p(s)$ for each state s, where the absorbing states are either sampled at the beginning (Quenched problem), or after each step (Annealed problem). In both versions, the goal of MC with traps ([46]) is to decide whether or not the reaching probability at a

(stochastic) trapping state is 1, when starting from a specific state, x.

The main challenge of MC with traps is to handle a (possibly infinite) countable state space and an infinite horizon. In contrasts, SafeZone is defined over a finite state space and a finite horizon. Handling the SafeZone via trapping states problem is pointless, as it is most likely to return a negative answer for finite setting.

Given a MC, a trivial exponential time algorithm to find its SafeZone is to enumerate over all possible subsets of states, and compute their safety (as done in lemma 3.70). In general, the main challenges of the SafeZone problem are computational and sample complexity minimization (we address both).

We highlight some additional differences between the two problems:

1. The trapping states problem mainly addresses infinite-sized input, and the goal is to **decide** whether some absorbing state is eventually reached, or not. In contrast, in the SafeZone problem, we consider the **algorithmic** problem of computing a subset of states from which the escape probability within the H steps is small using trajectory samples alone. As a result, we do not see how one of these problems could help solve the other.

2. Even if there are no absorbing states within the MC (thus the trapping probability is 0), the SafeZone problem is still challenging (in particular, the hardness result still stands).

3. The safety of a subset of states depends on the subset itself, which is selected by the algorithm. In contrast, trapping states are sampled from a known distribution and induce a probability over reaching some absorbing state.

4. Our main challenge is to efficiently find a "good" subset of states. Given a finite-horizon MC and a subset of states, Lemma $E.1$ computes the escape

probability from the subset. Computing this probability is not a significant challenge, unlike in the case of trapping states.

5. Finally, unlike the SafeZone problem, in trapping states problems, access to the MC model is assumed.

3.3.10 Proofs for Gentle Start (Section 3.3.3)

Greedy by Threshold Algorithm

A naive approach to the SafeZone problem is to return all states $s \in \mathcal{S}$ with probability $p(s) \geq \beta$, for some parameter $\beta > 0$, see Algorithm 14.

Algorithm 14 Greedy by Threshold

Parameter: $\beta > 0, \{p(s)\}_{s \in \mathcal{S}}$
return $\{s \in \mathcal{S} : p(s) \geq \beta\}$

Lemma 3.62. *For any $\rho, \beta \in (0,1)$, the Greedy by Threshold Algorithm returns a set that is $(\rho + k^*\beta, \frac{H}{\beta}) - $ SafeZone. In particular, for $\beta = \frac{\rho}{k^*}$, this set is $\left(2\rho, \frac{k^* H}{\rho}\right) - $ SafeZone.*

Proof. There are at most $\frac{H}{\beta}$ states with probability $p(s) \geq \beta$. Thus $|F| \leq \frac{H}{\beta}$.

Denote by F^* the optimal $(\rho, k^*) -$ SafeZone set. By law of total probability,

$$\Pr_{\tau}[\tau \not\subseteq F] \leq \Pr_{\tau}[\tau \not\subseteq F^*] + \Pr_{\tau}[\tau \subseteq F^* \setminus F].$$

Looking at the R.H.S of the inequality, the left term is smaller than ρ by the definition of SafeZone. The right term is equal to the probability to reach a state in F^* that its probability is smaller than β, i.e., a state in $F^* \setminus F$.

Using union bound, this can be bounded by $k^*\beta$. ∎

Lemma 3.71. *For every $\rho \in (0, 1/2), H \in \mathbb{N}$, there exists an MDP and a minimal integer k such that the MDP has a $(\rho, k)-$SafeZone , but for $\beta = \rho/k$ Greedy by Threshold Algorithm returns F with escape probability $\leq 2\rho$ and of size $|F| = \Omega(H/\beta)$.*

Proof. Fix $\rho \in (0, 1)$. For ease of the presentation we will assume that $\frac{1-\rho}{\beta}$ is an integer (if not, it should be rounded to the nearest integer). Define A to contain $\frac{1-\rho}{\beta} \cdot H$ states, B to contain $k-1$ states, and $\mathcal{S} = \{s_0\} \cup A \cup B$. Consider the following MDP with states $\mathcal{S}$ and starting state s_0. The transition function is defined as follows:

- For every $i \in A$, $\Pr[s^A_{1,i}|s_0] = \beta$ and for every $j \in [H-1]$, $\Pr[s^A_{j+1,i}|s^A_{j,i}] = 1$.
- For $s \in B$, $\Pr[s|s_0] = \frac{1-\rho}{k-1}$
- For $s \in B$, $\Pr[s|s] = 1$

The MDP is illustrated in fig. 3.8. Clearly, $\{s_0\} \cup B$ is a $(\rho, k)-$SafeZone . In addition, Greedy by Threshold Algorithm returns the set of all states, as for every state $s \in A$ we have that $p(s) = \beta$, $p(s_0) = 1 > \rho \geq \beta$, and for every $s \in B$ we have that $p(s) = \frac{1-\rho}{k-1} > \frac{\rho}{k} = \beta$. Thus the size of the returned set is $\mathcal{S}$, which is of size $\Omega(H/\beta)$, which completes the proof. ∎

Simulation Algorithm

Algorithm 15 Simulation Algorithm

Input: $m = \frac{1}{\beta} \ln \frac{k^*}{0.005}$
$F \leftarrow \{s_0\}$
for $i = 1 \dots m$ **do**
  $\tau \leftarrow$ choose a random trajectory
  $F \leftarrow F \cup \tau$
end for
return F

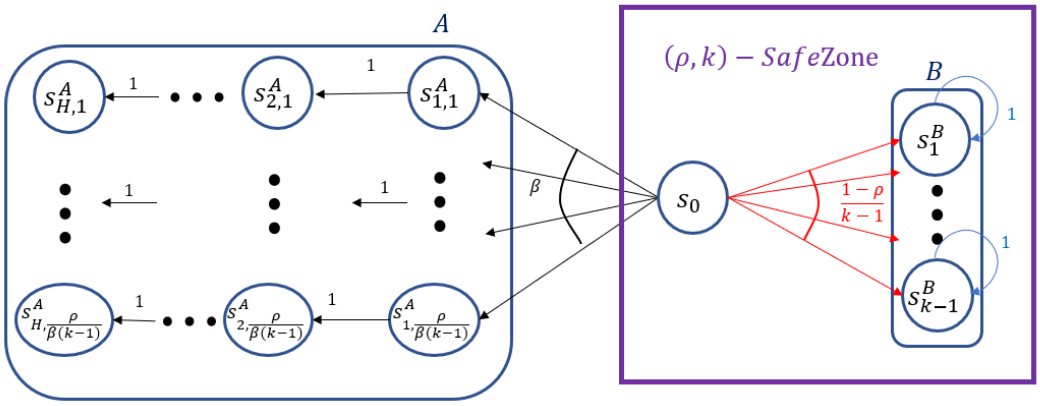


Figure 3.8: Lower bound for Greedy By Threshold Algorithm.

Lemma 3.63. *Fix $\rho, \beta \in (0,1)$. With probability at least 0.99, Simulation Algorithm returns a set that is $\left(\rho + k^*\beta, O(k^* + \frac{\rho H \ln k^*}{\beta})\right) - SafeZone$. In particular, for $\beta = \frac{\rho}{k^*}$, this set is $(2\rho, O(k^* H \ln k^*)) - SafeZone$.*

Proof. Denote by F^* the optimal $(\rho, k^*) - $ SafeZone set. By the law of total expectation, we can split $\mathbb{E}[|F|]$ into two parts, depending on whether trajectories are entirely in F^* or not:

- Trajectories that are entirely in F^* contribute at most k^* states to F.
- A trajectory that is not contained in F^* contributes at most H states to F.

Thus,

$$\mathbb{E}[|F|] \leq k^* + \rho \cdot \left(\frac{1}{\beta} \ln \frac{k^*}{0.005}\right) \cdot H = O\left(k^* + \frac{\rho H \ln k^*}{\beta}\right).$$

We use Markov's inequality to get the desired bound on $|F|$.

For the safety, we first denote the set of all states in F^* with probability at least β as

$\Gamma = \{s \in F^* \mid p(s) \geq \beta\}$. We will show that with probability at least 0.9995, it holds that $\Gamma \subseteq F$, which will prove our claim, similarly to Lemma 3.62.

For a fixed state $s \in \Gamma$, the probability that $s \notin F$ is bounded by $(1-p(s))^{\frac{1}{\beta}\ln\frac{k^*}{0.005}} \leq e^{-\frac{\beta}{\beta}\cdot\ln\frac{k^*}{0.005}} = \frac{0.005}{k^*}$. Using union bound, the probability that there is a state $s \in \Gamma$ which is not in F is bounded by $k^* \cdot \frac{0.005}{k^*} = 0.005$.

In other words, with probability at least 0.995, $\Gamma \subseteq F$, thus implementing the greedy approach in Algorithm 14 and proving that the probability that a random trajectory escapes F is bounded by $\rho + k^*\beta$. ∎

Lemma 3.72. *For every $\rho, \gamma \in (0,1)$, $H, k \in \mathbb{N}$, and $\beta = \frac{\rho}{k}$, there is an integer $r \in \mathbb{N}$ and MDP with $(\rho,k)-$SafeZone, but with probability $\geq 1-\gamma$, Simulation algorithm returns F of size $\mathbb{E}[|F|] \geq kH\ln k$ with escape probability $\Delta(F) = O(\rho)$.*

Proof. Fix $\rho, \gamma \in (0,1)$. Recall that $m = \frac{1}{\beta}\ln\frac{k^*}{0.005}$ and take $r = \lceil\frac{m^2}{\gamma}\rceil$. Define A to contain rH states, B to contain $k-1$ states, and $\mathcal{S} = \{s_0\} \cup A \cup B$.

Consider the following MDP with states $\mathcal{S}$ and starting state s_0. The transition function is defined as follows:

- For every $i \in A$, $\Pr[s^A_{1,i}|s_0] = \frac{\rho}{r}$ and for every $j \in [H-1]$, $\Pr[s^A_{j+1,i}|s^A_{j,i}] = 1$.
- For $s \in B$, $\Pr[s|s_0] = \frac{1-\rho}{k-1}$
- For $s \in B$, $\Pr[s|s] = 1$

The MDP is illustrated in Figure 3.9.

The set $B \cup \{s_0\}$ is $\rho-$safe with k states.

We will show that:

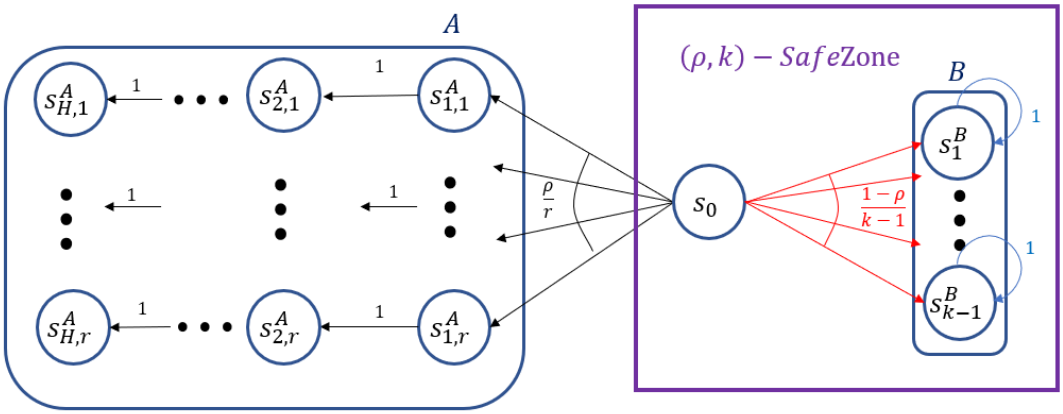


Figure 3.9: Lower bound for SIMULATION Algorithm.

- After adding $\geq \frac{1}{\beta} \ln k = \frac{k}{\rho} \ln k$ random trajectories, with probability $\geq 1 - \gamma$ we have that $|F| \geq kH \ln k$.
- After adding m random trajectories, we have that with high probability $F^* \subseteq F$, thus $\Delta(F) \leq \Omega(\rho)$.

To prove the first property, we claim that with probability$\geq 1 - \gamma$, every time we add a trajectory τ such that $\tau \cap A \neq \emptyset$, we add H new states.

Notice that if we ignore s_0, trajectories in A are entirely unconnected, and each trajectory is chosen randomly with probability $\Pr[s^A_{1,i}|s_0] = \frac{\rho}{r}$. This yields that if $s^A_{1,i} \notin F$, then $s^A_{j,i} \notin F$ for every $j \in [H]$. As a result, every time we add a new $s^A_{1,i}$ to F, we add $H - 1$ more states to F. Let N denotes the amount of trajectories sampled with states from A. The probability that their intersection contains only s_0 is

$$\frac{r \cdot (r-1) \cdot \ldots \cdot (r-N)}{r^N} \geq \left(\frac{r-N}{r}\right)^N = \left(1 - \frac{N}{r}\right)^N \geq 1 - \frac{N^2}{r} = 1 - \gamma.$$

From the structure of the MDP, we have that $\mathbb{E}[N] = \rho m$. Therefore, with probability

$\geq 1 - \gamma$,

$$\mathbb{E}[|F|] \geq \mathbb{E}[N] \cdot H = \rho \cdot m \cdot H \geq \rho \cdot \frac{1}{\beta} \ln k \cdot H = kH \ln k.$$

The second property follows from Lemma 3.63.

■

Greedy at Each Step

Algorithm 16 Greedy at Each Step

Input: $\rho > 0, \{p(s)\}_{s \in \mathcal{S}}$
$F \leftarrow \{s_0\}$
for $i = 1 \dots H$ **do**
  Sort states in $\mathcal{S}_i$, $p(s_i^1) \geq \dots \geq p(s_i^{|\mathcal{S}_i|})$
  $j^* \leftarrow \text{argmin}_{j \in [|\mathcal{S}_i|]} \sum_{r=1}^{j} p(s_i^r) \geq 1 - \rho$
  $F \leftarrow F \cup \left\{ s_i^1, \dots s_i^{j^*} \right\}$
end for
return F

Lemma 3.64. *For any $\rho \in (0, 1)$, if the MDP is layered, Greedy at Each Step Algorithm returns a set that is $(\rho H, k^*) - SafeZone$.*

Proof. Take a random trajectory $\tau = (s_1, s_2, \dots)$. For every $s_i \in \tau$, the probability that $s_i \notin F$ is bounded by ρ, thus using union bound, the probability that τ has state s_i such that $s_i \notin F$ is at most ρH.

The construction of F guarantees that F is the minimal subset of states such that for every i, the probability that s_i is in the subset is at least $1 - \rho$. Assume by contradiction that $|F| > k^*$. Then there is a time step i such that $\Pr[s_i \in F^*] < 1 - \rho$, which is a contradiction, since $\Pr[\tau \in F^*] \leq \min_i \Pr[s_i \in F^*]$.

■

Lemma 3.73. *For any $\rho \in (0,1)$, there is an MDP and an integer k such that there is a $(\rho, k)-$*SafeZone *, but* Greedy At Each Step *Algorithm returns F with escape probability $\Delta(F) \geq \Omega(H\rho)$.*

Proof. Fix $\rho \in (0,1)$ and take $k = 3H + 1$.

Consider the MDP illustrated in Figure 3.10. The set $\{s_0\} \cup \{s_1^i\}_i \cup \{s_2^i\}_i \cup \{s_3^i\}_i$ form a $(\rho, 3H+1)-$SafeZone .

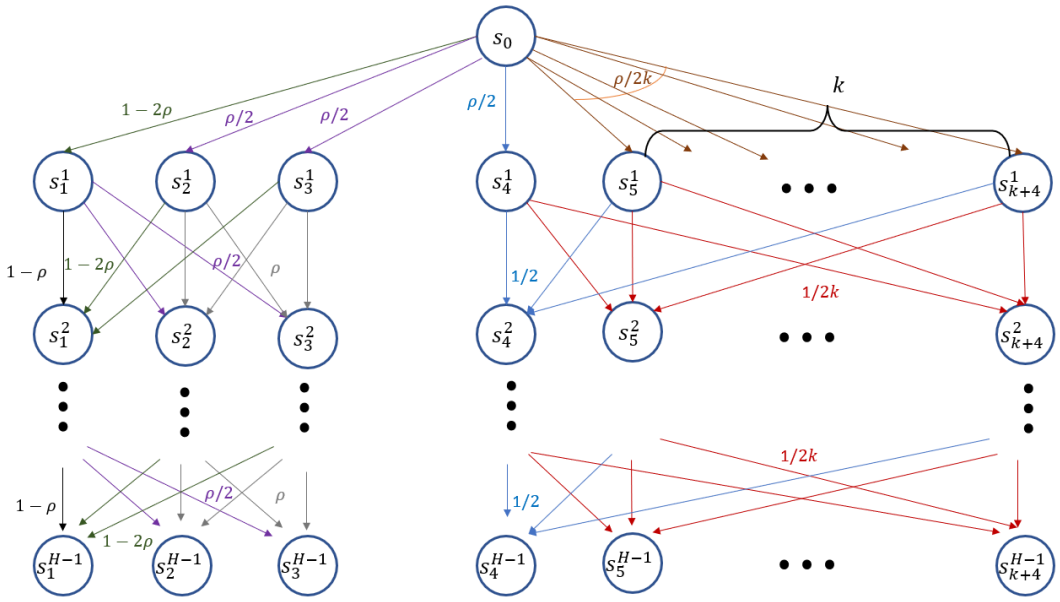


Figure 3.10: Lower bound for Greedy at Each Step Algorithm.

We will prove by induction that the for every time i,

- $p(s_1^i) = 1 - 2\rho$,
- $p(s_2^i) = p(s_3^i) = p(s_4^i) = \frac{\rho}{2}$, and
- For every $j \in \{5, \ldots, k+4\}$, $p(s_j^i) = \frac{\rho}{2k}$.

It is easy to see that the two properties hold for $i = 1$.

For $i > 1$,

$$p(s_1^i) = p(s_1^{i-1})(1-\rho) + p(s_2^{i-1})\frac{\rho}{2} + p(s_3^{i-1})\frac{\rho}{2} = (1-2\rho)(1-\rho) + 2(1-2\rho)\frac{\rho}{2} = 1-2\rho$$

$$p(s_2^i) = p_{i-1}(s_1^{i-1})\frac{\rho}{2} + p(s_2^{i-1})\rho + p(s_3^{i-1})\rho = (1-2\rho)\frac{\rho}{2} + \frac{\rho^2}{2} + \frac{\rho^2}{2} = \frac{\rho}{2}$$

Similarly, $p(s_3^i) = \frac{\rho}{2}$.

$$p(s_4^i) = \frac{1}{2}p(s_4^{i-1}) + \sum_{j=5}^{k+4}\frac{p(s_j^{i-1})}{2} = \frac{\rho}{4} + k\frac{\rho}{4k} = \frac{\rho}{2}$$

For every $j \in \{5, \ldots, k+4\}$,

$$p(s_j^i) = \frac{1}{2k}p(s_4^{i-1}) + \sum_{m=5}^{k+4}\frac{p(s_m^{i-1})}{2k} = \frac{\rho}{4k} + k\frac{\rho}{4k^2} = \frac{\rho}{2k}.$$

The algorithm might return $\{s_0\} \cup \{s_1^i\}_i \cup \{s_2^i\}_i \cup \{s_4^i\}_i$, i.e., instead of taking $\cup_i\{s_3^i\}_i$ it takes $\cup_i\{s_4^i\}_i$. Finally, the observation $\Delta(\{s_0\} \cup \{s_1^i\}_i \cup \{s_2^i\}_i \cup \{s_4^i\}_i) \geq \frac{\rho H}{4}$ completes the proof. ■

3.3.11 Algorithm for Detecting SafeZones: Full Analysis (Section 3.3.4)

For convince, we state here Hoeffding's inequality.

Lemma 3.74. *[Hoeffding's Inequality] Let $y_1, \ldots, y_N$ be independent random variables such that $y_i \in [a, b]$ for every y_i with probability 1. Then, for any $\epsilon > 0$,*

$$\Pr\left[\left|\frac{1}{N}\sum_{i=1}^{N} y_i - \mathbb{E}[y_i]\right| \geq \epsilon\right] \leq 2e^{-2N\epsilon^2/(b-a)^2}.$$

Proof of theorem 3.65

In this section we provide a complete proof for theorem 3.65. Throughout the section, we define a few terms and notions. We will start with proving guarantees regarding a single iteration of the while–loop.

Recall that F^* denotes a minimal $\rho-$safe set (of size k^*). If there are multiple optimal solutions, choose one arbitrarily. For the convince of analysis, we denote the values of the algorithm variables at the end of each iteration i of the while–loop by $\tau_i, F_i, accept_i$. Let $j(i)$ denote the value of variable j during the $i-$th call to *EstSafety* Subroutine. In addition, let N_i denote the number of trajectories sampled for the $j-$th time of calling Subroutine *EstSafety*, i.e., $N_i = \frac{1}{2\epsilon^2} \ln \frac{2}{\lambda_{j(i)}}$ for $j(i) \leq i$.

For ease of presentation, we recall some of the definitions from the proof technique description. We say that a trajectory τ is *good* if all the states in τ are in F^* and *bad* if it escapes it. I.e., a trajectory is good if $\tau \subseteq F^*$ and bad if $\tau \not\subseteq F^*$. Additionally, we say that a state $s \in \mathcal{S}$ is *good* if it is in F^* and *bad* otherwise. Namely, a state s is good if $s \in F^*$ and bad if $s \notin F^*$. Let $G_i(F_{i-1})$ and $B_i(F_{i-1})$ be the number of good and bad states added to F_{i-1} in iteration i, respectively (notice that $G_i(F_{i-1})$ and $B_i(F_{i-1})$ are random variables that depends on F_{i-1}). For short, whenever it is clear from the context, we write G_i and B_i respectively.

The following lemma bounds the error in approximating the escape probability.

Lemma 3.75. *Let $F_{i-1} \subseteq \mathcal{S}$ be a subset of of states and $\epsilon, \lambda_j > 0$ be some parameters. Let S_i be a sample of $N_i \geq \frac{1}{2\epsilon^2} \ln \frac{2}{\lambda_{j(i)}}$ i.i.d. random trajectories. Then,*

$$\Pr_{S_i}\left[\left|\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})\right| \geq \epsilon\right] \leq \lambda_j.$$

Also, as $\lambda_j = \frac{3\lambda}{2(\pi j)^2}$,

$$\Pr\left[\exists i \ \left|\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})\right| \geq \epsilon\right] \leq \lambda/4,$$

Where the last probability is over all the samples S_i *made by EstSafety Subroutine.*

Proof. The first part follows directly from Hoeffding's inequality by taking $y_i = \mathbb{I}[\tau \not\subseteq F]$.

Assigning $\lambda_j = \frac{3\lambda}{2(\pi j)^2}$ and applying union bound, we get

$$\begin{aligned}\Pr\left[\exists i \ \left|\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})\right| \geq \epsilon\right] &\leq \sum_i \Pr_{S_i}\left[\left|\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})\right| \geq \epsilon\right] \\ &\leq_{(*)} \sum_{j(i)} \lambda_{j(i)} \leq \sum_{j=1}^{\infty} \lambda_j = \sum_{j=1}^{\infty} \frac{3\lambda}{2(\pi j)^2} = \frac{\lambda}{4}.\end{aligned}$$

The inequality marked by $(*)$ follows from the fact that $\Delta(F)$ is estimated once for every time j increases. ∎

We define the event that *EstSafety* always provides good estimations by

$$\mathcal{E} = \{\forall i \ \left|\widehat{\Delta}(F_{i-1}) - \Delta(F_{i-1})\right| \leq \epsilon\}.$$

By the above we have that $\Pr[\mathcal{E}] \geq 1 - \lambda/4$.

In the following lemma we assume that if the current escape probability is at least 2ρ, then the fraction of bad trajectories that escape F_{i-1} is bounded from above by the fraction of good trajectories that escape F_{i-1}.

Lemma 3.76. *Let* $\rho > 0$ *and assume that* $\Delta(F_{i-1}) \geq 2\rho$. *Then,*

$$\Pr_{\tau}[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \not\subseteq F^*] \leq \Pr_{\tau}[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*],$$

where the probabilities are over random trajectories.

Proof. To prove the lemma, we will bound the probability $\Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \not\subseteq F^*]$ from above and the probability $\Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*]$ from below. Since $\Delta(F^*) \leq \rho$,

$$\Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \not\subseteq F^*] \leq \Pr_\tau[\tau \not\subseteq F^*] \leq \rho. \tag{3.36}$$

The assumption $\Delta(F_{i-1}) \geq 2\rho$ implies that

$$\begin{aligned} 2\rho \leq \Delta(F_{i-1}) &= \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0] = \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*] + \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \not\subseteq F^*] \\ &\leq \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*] + \Pr_\tau[\tau \not\subseteq F^*] \leq \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*] + \rho, \end{aligned}$$

hence

$$\rho \leq \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*]. \tag{3.37}$$

Putting (3.36) and (3.37) together yields the statement. ∎

Now, as long as the algorithm is inside the while–loop (i.e., the escape probability holds $\widehat{\Delta}(F) > 2\rho + \epsilon$), it follows that $\Delta(F) \geq 2\rho$ with high probability from lemma 3.75. Combining it with lemma 3.76 would yield that with high probability over a random trajectory, if the trajectory escapes F then in expectation it is at least as likely to be good as it is to be bad.

We move on to show the main ingredient of the proof, namely that for any iteration, with high probability, the expected number of good states added to the current set F is larger or equal to the expected number of bad states.

For every iteration i in which we sample τ_i both G_i and B_i depends on the following:

1. The realizations of the sampled trajectory, τ_i, and in particular on $new_{F_{i-1}}(\tau_i)$.

2. The probability of adding it to F, i.e., $1/new_{F_{i-1}}(\tau_i)$.

Next, we prove (3.35).

Lemma 3.77. *Assume event $\mathcal{E}$ holds. Thus, for all iterations i inside the while–loop we have*

$$\mathbb{E}[B_i|F_{i-1}] \leq \mathbb{E}[G_i|F_{i-1}],$$

where the expectation is over the trajectory τ that is sampled from the MC dynamics and added to F_{i-1} according to $Q_{F_{i-1}}$.

Proof. Since event $\mathcal{E}$ holds, we have that $\Delta(F_{i-1}) \geq 2\rho$ as long as we do not terminate in iteration i.

We can use it to bound $\mathbb{E}_\tau[B_i|F_{i-1}]$ by

$$\begin{aligned}
\mathbb{E}_\tau[B_i|F_{i-1}] &\leq \sum_{h=1}^{H} \frac{\Pr_\tau[new_{F_{i-1}}(\tau) = h \wedge \tau \not\subseteq F^*]}{h} \cdot h \\
&= \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \not\subseteq F^*] \underbrace{\leq}_{lemma\ 3.76} \Pr_\tau[new_{F_{i-1}}(\tau) \neq 0 \wedge \tau \subseteq F^*] \\
&= \sum_{h=1}^{H} \frac{\Pr_\tau[new_{F_{i-1}}(\tau) = h \wedge \tau \subseteq F^*]}{h} \cdot h \leq \mathbb{E}_\tau[G_i|F_{i-1}].
\end{aligned}$$

■

Theorem 3.65. *Given $\rho, \epsilon, \lambda \in (0,1)$, FINDING SAFEZONE Algorithm returns a subset $F \subseteq \mathcal{S}$ such that:*

1. *The escape probability is bounded from above by $\Delta(F) \leq 2\rho + 2\epsilon$, with probability $1 - \lambda$.*
2. *The expected size of F given $\mathcal{E}$ is bounded by $\mathbb{E}[|F| \mid \mathcal{E}] \leq 2k^*$.*

3. *The sample complexity of the algorithm is bounded by* $O\left(\frac{k^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda}+\frac{Hk^*}{\rho\lambda}\right)$*, and the running time is bounded by* $O\left(\frac{Hk^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda}+\frac{H^2k^*}{\rho\lambda}\right)$*, with probability* $1-\lambda$*.*

Proof. Assume that the event $\mathcal{E}$ holds, and recall that

$$\Pr[\mathcal{E}] \geq 1-\lambda/4. \tag{3.38}$$

We start with the first clause. Since event $\mathcal{E}$ holds, lemma 3.75 in particular implies that $\Delta(F) \leq 2\rho + 2\epsilon$, hence the first clause holds.

For second clause, we will bound $\mathbb{E}[|F| \mid \mathcal{E}]$ from above by $2k^*$. Since $\mathcal{E}$ holds, we have that $\Delta(F_{i-1}) \geq 2\rho$, for every i inside the while–loop, thus Lemma 3.77 yields

$$\mathbb{E}[B_i|F_{i-1}] \leq \mathbb{E}[G_i|F_{i-1}].$$

This implies that

$$\mathbb{E}[|F| \mid \mathcal{E}] \leq 2\sum_i \mathbb{E}_{F_{i-1}}[\mathbb{E}[G_i|F_{i-1}]]|\mathcal{E}] \leq 2k^*, \tag{3.39}$$

where the last inequality follows from the definition of G_i, as $\sum_i G_i \leq |F^*| = k^*$.

We continue with the third clause of the theorem. Let M denote the sample complexity of the algorithm, namely $M = M_F + M_E$ where M_F is the expected total number of trajectories sampled within the FINDING SAFEZONE Algorithm (without the samples made by *EstSafety* Subroutine) and M_E is total number of trajectories sampled using *EstSafety*. We will bound each term separately.

Since $\mathcal{E}$ holds, whenever we are inside the while–loop, $\Delta(F_i) \geq 2\rho$, which implies that it takes at most $1/2\rho$ trajectories in expectation to sample a trajectory that escapes

F_i, and such trajectory is accepted with probability at least $1/H$. Thus, from Wald's identity, it follows that

$$\mathbb{E}\left[M_F \middle| \mathcal{E}\right] = \frac{H}{2\rho} \cdot \mathbb{E}[|F| \ |\mathcal{E}] \leq \frac{Hk^*}{\rho}.$$

From Markov's inequality on the above inequality, with probability at least $1 - \frac{\lambda}{4}$,

$$\Pr\left[M_F \geq \frac{4Hk^*}{\rho\lambda} \middle| \mathcal{E}\right] \leq \frac{\lambda}{4}. \tag{3.40}$$

Moving on to bound M_E. Since $\mathcal{E}$ holds, it follows from (3.39) and Markov's inequality that

$$\Pr\left[|F| \geq \frac{8k^*}{\lambda} \ \middle| \ \mathcal{E}\right] = \Pr\left[|F| \geq 2k^* \cdot \frac{4}{\lambda} \ \middle| \ \mathcal{E}\right] = \Pr\left[|F| \geq \mathbb{E}[|F| \mid \mathcal{E}] \cdot \frac{4}{\lambda} \ \middle| \ \mathcal{E}\right] \leq \frac{\lambda}{4}. \tag{3.41}$$

If $|F| \leq \frac{8k^*}{\lambda}$, the number of calls for Subroutine *EstSafety* is also bounded by $8\pi k^*/\lambda$ (we only call *EstSafety* after we added states to F). It also implies that $\frac{3\lambda^3}{2(8\pi k^*)^2} \leq \lambda_j$ for every $j \geq 1$. Thus, if $|F| \leq \frac{8k^*}{\lambda}$,

$$\begin{aligned} M_E = \sum_{j=1}^{|F|} N_i &\leq \sum_{j}^{\frac{8k^*}{\lambda}} \frac{1}{2\epsilon^2} \ln \frac{2}{\lambda_j} \leq \sum_{j}^{\frac{8k^*}{\lambda}} \frac{1}{2\epsilon^2} \ln \frac{2}{\frac{3\lambda^3}{2(8\pi k^*)^2}} \leq \sum_{j}^{\frac{8k^*}{\lambda}} \frac{1}{2\epsilon^2} \ln \frac{86(\pi k^*)^2}{\lambda^3} \\ &= \frac{8k^*}{2\lambda\epsilon^2} \ln \frac{86(\pi k^*)^2}{\lambda^3} = \frac{4k^*}{\lambda\epsilon^2} \ln \frac{86(\pi k^*)^2}{\lambda^3} \end{aligned}$$

Combining the above with (3.41), we get

$$\Pr\left[M_E > \frac{4k^*}{\lambda\epsilon^2} \ln \frac{86(\pi k^*)^2}{\lambda^3} \ \middle| \ \mathcal{E}\right] \leq \frac{\lambda}{4} \tag{3.42}$$

As $M = M_F + M_E$, union bound over (3.38), (3.40) and (3.42) implies that with

probability $\geq 1 - 3\lambda/4 > 1 - \lambda$,

$$M = O\left(\frac{k^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda} + \frac{Hk^*}{\rho\lambda}\right) \tag{3.43}$$

For each trajectory we sample we run in time $O(H)$, e.g., by using a lookup table for maintaining the current set F. Consequently, if the event in (3.43) holds then the running time of the algorithm is bounded by

$$O\left(\frac{Hk^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda} + \frac{H^2k^*}{\rho\lambda}\right).$$

Overall, all the clauses in the lemma hold with probability $\geq 1 - \lambda$.

■

Proof of Theorem 3.66

Theorem 3.66. *(main theorem) Given $\epsilon, \rho, \delta > 0$, if we run* FINDING SAFEZONE *for $\Theta(\frac{1}{\delta})$ times and return the smallest output set, $F \subseteq \mathcal{S}$, then with probability ≥ 0.99*

1. *The escape probability is bounded by $\Delta(F) \leq 2\rho + 2\epsilon$.*
2. *The size of F is bounded from above by $|F| \leq (2+\delta)k^*$.*
3. *The total sample complexity and running time are bounded by $O(\frac{k^*}{\delta^2\epsilon^2}\ln\frac{k^*}{\delta} + \frac{Hk^*}{\rho\delta^2})$, and $O(\frac{Hk^*}{\delta^2\epsilon^2}\ln\frac{k^*}{\delta} + \frac{H^2k^*}{\rho\delta^2})$, respectively.*

Proof. Assume we run FINDING SAFEZONE Algorithm for $m = \frac{2\ln 300}{\delta}$ times and denote each algorithm output by F^i. Return the smallest set $F = \text{argmin}_{F^i}|F^i|$.

It follows from theorem 3.65 that for every $\lambda \in (0,1)$, each F^i is of expected size $\mathbb{E}[|F^i|] \leq 2k^*$, and is $(2\rho + 2\epsilon)-$safe with probability $\geq 1 - \lambda$. Choosing $\lambda = \frac{0.01}{3m}$

implies

$$\Pr[\Delta(F) > 2\rho + 2\epsilon] \leq \frac{0.01}{3}. \tag{3.44}$$

In addition, from Markov's inequality it follows that for every $\delta > 0$,

$$\begin{aligned}
\Pr\left[|F^i| > (2+\delta)k^*\right] &\leq \Pr\left[|F^i| > (2+\delta)k^* | \mathcal{E}\right] + \Pr[\mathcal{E}] \\
&\leq \frac{2k^*}{(2+\delta)k^*} + \lambda \\
&= 1 - \frac{\delta/2}{1+\delta/2} + \lambda \\
&= 1 - \frac{\delta/2 - \lambda - \lambda\delta/2}{1+\delta/2}
\end{aligned}$$

From the independence of the algorithm runs, for $m = \frac{2\ln 300}{\delta}$,

$$\begin{aligned}
\Pr[|F| > (2+\delta)k^*] &\leq \Pr[\forall i : (|F^i| > (2+\delta)k^*)] \\
&\leq \prod_{i\in[m]} \Pr[|F^i| > (2+\delta)k^*] \\
&\leq \left(1 - \frac{\delta/2 - \lambda - \lambda\delta/2}{1+\delta/2}\right)^m \\
&\leq e^{-m(\frac{\delta/2-\lambda-\lambda\delta/2}{1+\delta/2})} \leq \frac{0.01}{3}.
\end{aligned}$$

Hence

$$\Pr[|F| > (2+\delta)k^*] \leq \frac{0.01}{3}. \tag{3.45}$$

As for the sample complexity, let M_i denote the (random) sample complexity of the

$i-$th run, and let us denote

$$\bar{M} = \frac{4k^*}{\lambda\epsilon^2}\ln\frac{86(\pi k^*)^2}{\lambda^3} + \frac{4Hk^*}{\rho\lambda}.$$

From theorem 3.65, $M_i > \bar{M}$ with probability $< \lambda$.

By taking union bound on the sample complexity bound per one run, we get

$$\Pr\left[\exists i : M_i > \bar{M}\right] \leq \sum_{i\in[m]} \Pr\left[M_i > \bar{M}\right] \leq m \cdot \lambda = \frac{0.01}{3}.$$

Where the last inequality follows from theorem 3.65, and $\lambda = \frac{0.01}{3m}$.

Assigning $m = \frac{2\ln 300}{\delta}$ and $\lambda = \frac{0.01}{3m} = \frac{0.01\delta}{6\ln 300}$, we get that with probability $\geq 1 - \frac{0.01}{3}$,

$$\sum_{i=1}^{m} M_i = O\left(\frac{mk^*}{\lambda\epsilon^2}\ln\frac{k^*}{\lambda} + \frac{mHk^*}{\rho\lambda}\right) = O\left(\frac{k^*}{\delta^2\epsilon^2}\ln\frac{k^*}{\delta} + \frac{Hk^*}{\rho\delta^2}\right) \tag{3.46}$$

Since the algorithm runs in time $O(H)$ for every trajectory sampled, if the sample complexity is bounded by the above term, then the total running time is bounded by $O\left(\frac{Hk^*}{\delta^2\epsilon^2}\ln\frac{Hk^*}{\delta} + \frac{Hk^*}{\rho\delta^2}\right)$.
Finally, from union bound over (3.44), (3.45) and (3.46) all the theorem properties hold with probability ≥ 0.99. ∎

3.3.12 Hardness Proofs (Section 3.3.5)

Theorem 3.68. *For every graph $G = (V, E)$ and an integer k_c there exists a clique of size k_c in $G \iff$ SafeZone$(M(G), k_c, \rho)$ answers YES.*

Proof. ($\Longrightarrow$) If there is a clique of size k_c, then we can take the corresponding k states. The probability to remain in this subset is at least $\left(\frac{k-1}{d}\right)^2$. Thus, an exact solver for SafeZone must return YES.

($\Longleftarrow$) Suppose there is no clique of size k. Assume by contradiction that the reduction (algorithm) returns YES. Let s_0 be a vertex which was the starting state from the running instance which the YES came from and let $\hat{F}$ denote the output of SafeZone . We will show that the probability to remain in any subset of size k is smaller than $\left(\frac{k-1}{d}\right)^2$.

Since there is no clique of size k in G, we know that $\hat{F}$ is not a clique. It therefore follows that there exists at least two vertexes, $s_a, s_b \in V$ such that $(s_a, s_b) \notin E$.

We will now bound the probability of escape from state s_0 by exhaustion.

1. If $s_0 \neq s_a$, then

$$\Pr[escape\ from\ s_0] \geq \Pr[t = 1 : (s_0, s'), s' \notin \hat{F}]$$

$$+ \Pr[t = 1 : (s_0, s), s \neq s_a] \cdot \Pr[t = 2 : (s, s'), s' \notin \hat{F} | t = 1 : (s_0, s), s \neq s_a]$$

$$+ \Pr[t = 1 : (s_0, s_a)] \cdot \Pr[t = 2 : (s_a, s'), s' \notin \hat{F} | t = 1 : (s_0, s_a)]$$

$$= \frac{d-(k-1)}{d} + \frac{k-2}{d} \cdot \frac{d-(k-1)}{d} + \frac{1}{d} \cdot \frac{d-(k-2)}{d}$$

$$= 1 - \frac{k-1}{d} + \frac{k-2}{d} - \frac{(k-2)(k-1)}{d^2} + \frac{1}{d} - \frac{k-2}{d^2} =$$

$$1 - \frac{k-2}{d^2}(k-1+1) = 1 - \frac{k(k-2)}{d^2}$$

Hence

$$\Pr[staying] \leq \frac{k(k-2)}{d^2} < \frac{(k-1)^2}{d^2}.$$

2. If $s_0 = s_a$, then

$$\Pr[escape\ from\ s_0] \geq \Pr[t = 1 : (s_0, s'), s' \notin \hat{F}]$$

$$+ \Pr[t = 1 : (s_0, s), s \in \hat{F}] \cdot \Pr[t = 2 : (s, s'), s' \notin \hat{F} | t = 1 : (s_0, s), s \in \hat{F}]$$

$$= \frac{d - (k-2)}{d} + \frac{k-2}{d} \cdot \frac{d - (k-1)}{d}$$

$$= 1 - \frac{k-2}{d} + \frac{k-2}{d} - \frac{(k-2)(k-1)}{d^2}$$

$$= 1 - \frac{(k-2)(k-1)}{d^2}$$

Hence

$$\Pr[staying] \leq \frac{(k-2)(k-1)}{d^2} < \frac{(k-1)^2}{d^2}.$$

■

Chapter 4

Societal Challenges

Introduction

When we deal with decision-making that concerns individual people, ensuring societal requirements in data-driven algorithms such as fairness or safety is a key ingredient in making ML trustworthy thus making the practice of ML more acceptable. *Fairness* is a societal concern that addresses potential discrimination by algorithms of individual people or subgroup (of population) described by a *protected feature* - race, gender, disability, etc. There is a wide variety of fairness definitions and it became a major research area in ML (see, [13, 80, 99]).

Efficient Candidate Screening under Multiple Tests and Implications for Fairness In a paper published in FORC 2020 [39], we analyzed how a complex hiring process interacts with the requirements of fairness. When recruiting job candidates, employers rarely observe their underlying skill level directly. Instead, they must administer a series of interviews and/or collate other noisy signals in order to estimate the worker's skill. Traditional economics papers address screening models where employers access

worker skill via a single noisy signal. In this paper, we extend this theoretical analysis to a multi-test setting, considering both Bernoulli and Gaussian models. We analyze the optimal employer policy both when the employer sets a fixed number of tests per candidate and when the employer can set a dynamic policy, assigning further tests adaptively based on results from the previous tests. To start, we characterize the optimal policy when employees constitute a single group, demonstrating some interesting trade-offs. Subsequently, we address the multi-group setting, demonstrating that when the noise levels vary across groups, a fundamental impossibility emerges whereby we cannot administer the same number of tests, subject candidates to the same decision rule, and yet realize the same outcomes in both groups. We consider the ramifications for fairness within our model when employees, despite possessing similarly-distributed skills, are evaluated with different noise levels. We show impossibility results as well as a solution to equalize confusion matrix entries by adjusting the number of tests according to group parameters. Finally, we present a simple way to estimate group parameters without knowing the true skill levels (i.e., unsupervised learning), and give bounds in terms of the number of candidates.

Biased algorithms can reinforce existing inequalities and create new ones. If certain groups are consistently disadvantaged by screening systems, it can have significant social consequences, leading to reduced opportunities, discrimination, and other harms. In the final chapter of this thesis, we address this important topic through the eyes of optimization and ML toolbox. We suggest efficient candidate screening solutions and make sure they involve the consideration of different populations with different characteristics while optimizing the system's performance.

4.1 Candidate Screening and Implications for Fairness

4.1.1 Introduction

Consider an employer seeking to hire new employees. Clearly, the employer would like to hire the best employees for the task, but how will she know which are best fit? Typically, the employee will gather information on each candidate, including their education, work history, reference letters, and for many jobs, they will actively conduct interviews. Altogether, this information can be viewed as the *signal* available to the employer.

Suppose that candidates can be either *skilled* or *unskilled*. If the firm hires an "unskilled" candidate, it will incur a significant cost on account of lost productivity. For this reason, the employer would like to minimize the number of *False Positive* mistakes, instances where *unskilled* candidates are hired. On the other hand, the employer desires not to *overspend* on the hiring process, limiting the number of interviews per hired candidate (either on average, or absolutely). However, fewer interviews weakens the signal, causing the employer to make more mistakes. At the heart of our model is this inherent trade-off between the quality of the signal and the cost of obtaining the signal. This marks a departure from the classical economics literature, in which the signal is commonly regarded as a given.

Complicating matters, hiring efficiency is not the only desiderata at play. In society, candidates belong to various *demographic groups*, and we may strive to design policies that are in some sense *fair* vis-a-vis group membership. While *fairness* can be an elusive notion, regulators must translate it to concrete rules and laws. In the United States, a body of anti-discrimination law dating to the Civil Rights act of 1964, subjects decisions that result in disparate outcomes (as delineated by race, age, gender, religion, etc.) to extra scrutiny: employers must not only show that

preference for some groups over others did not drive the decision (disparate treatment doctrine) but also justify that any observed disparities arise from a business necessity (disparate impact doctrine), whether or not those disparities were intentional.

In this section, we seek to understand how a complex hiring process would interact with the requirements of fairness. We extend the theory on candidate screening and statistical discrimination, addressing the setting in which employers can subject employees to multiple tests, which we assume to be conditionally independent given the worker's skill level and group membership. To build intuition, most of our analysis focuses on a Bernoulli model of both worker skill and screening. Additionally, we extend the traditionally-studied Gaussian skill and screening models to the multi-test setting (Section 4.1.5).

Unlike the classical papers, in which an employer's hiring policy is given by a simple thresholding rule, our setting requires greater care to derive the optimal employer policy. In our setting, we imagine that the employer wishes to minimize the number of tests performed subject to a constraint upper-bounding the false positive rate. We characterize the optimal policy in this case as a randomized threshold policy. Subsequently, we show that this is not always an optimal policy and consider the setting in which employers can allocate tests dynamically. Namely, employers decide after each result whether to (i) hire the candidate; (ii) reject the candidate and move on to the next one; or (iii) administer a subsequent test. In the Bernoulli case, the optimal policy consists of administering tests until each candidate's posterior likelihood of being a high-skilled worker either dips below the prior or rises above a threshold determined by the tolerable false positive rate. We reduce the analysis of this process to a random walk over the log posterior odds and derive the solution via the corresponding Gambler's ruin problem.

we consider the ramifications for fairness within our model when employees, despite

possessing similarly-distributed skills, are evaluated with differing noise levels. We show impossibility results, as well as, a solution to equalize confusion matrix entries by adjusting the number of tests according to group parameters. Finally, we present a simple way to estimate group parameters without knowing the true skill levels (i.e., unsupervised learning), and give bounds in terms of the number of candidates from a group for good estimation.

Related work

The classical economics literature on discrimination in employment can broadly be divided into two focuses. The *taste-based discrimination* model due to [15] models the market outcomes in a setting where employers express an explicit preference for hiring members of one group, acting as if an employee's demographic membership provides utility. This preference for certain groups induces a sorting of employees from the disadvantaged group towards those employers who discriminate the least with wages ultimately determined by the *marginal discriminator*. Subsequently, [107] suggested a statistical mechanism by which similarly-skilled employees from different groups might experience differential outcomes: the comparative difficulty of screening from one group vs. another. Many subsequent works extend this analysis, typically focusing on Gaussian models of worker quality and conditionally-Gaussian test scores [6, 2]. These papers consider the setting where workers are assessed via a single test characterized by a group-dependent noise level. Our work is differentiated from these by considering richer mechanisms for acquiring signal.

In the more recent literature on fairness in machine learning, researchers often focus on binary classification, with employees characterized by a protected characteristic (group membership), and other (non-protected) covariates [104, 76, 77]. There, the predictor is presumably used to guide a consequential decision, such as allocating

some economic good (loans, jobs, etc.) [43] or assessing some penalty (e.g. risk scores to guide bail decisions) [36]. Papers then focus on various interventions for ensuring accurate prediction subject to various constraints such as demographic parity (outcomes independent of group membership), blindness (model cannot observe group membership), and equalized false negative and/or false positive rates [63]. Several simple impossibility results preclude simultaneously satisfying several combinations of these parities [20, 36, 84]. More recently, a number of papers have drawn inspiration from economic modeling, extending the literature on fairness in classification to consider longer-term dynamics, equilibria, and the emergence of feedback loops [69, 63, 50]. Finally, [12, 126] provide a survey of definitions from the algorithmic fairness literature.

Unrelated to fairness, [116] consider a model that is somehow resembles to ours in the context of A/B testing. They minimize the expected time per discovery (which can be viewed as hire) from an infinite pool of hypotheses (which can be viewed as candidates) with a bounded false discovery rate.

4.1.2 Candidate Screening: The Bernoulli Model

We formalize our problem as follows. An employer accesses an infinite pool of candidates (indexed by $i \in \mathbb{N}^+$), each of which has some (hidden) *skill level* $y_i \in \{0, +1\}$, which denote *unskilled* and *skilled*, respectively. Underlying worker skill levels y_i are sampled independently from a Bernoulli distribution with parameter p. An employer can access information about the i-th candidate through a sequence of τ tests, which are conditionally independent given y_i. Each *test result*, $\hat{y}_{i,j} \in \{0, +1\}$ disagrees with the ground truth skill with probability $\Pr[\hat{y}_{i,j} \neq y_i] = \frac{1-\sigma}{2}$, where $\sigma \in (0, 1)$, i.e., $\hat{y}_{i,j} = y_i \oplus Br(\frac{1-\sigma}{2})$[1]. For convenience, we denote the noise level as

[1] $\oplus$ is the XOR operation between two binary random variables, and therefore $\hat{y}_{i,j}$ is also a random variable.

$\eta = \frac{1-\sigma}{2} \in (0, \frac{1}{2})$. We say that a test result $\hat{y}_{i,j}$ is *flipped* if $\hat{y}_{i,j} \neq y_i$, and the number of flipped results for a given candidate is denoted by Z_τ^η is $Z_\tau^\eta = \sum_{j=1}^{\tau} \mathbb{I}(\hat{y}_{i,j} \neq y_i)$, where $\mathbb{I}(\cdot)$ is the indicator function.

The employer decides weather or not to hire the current candidate, but unlike the secretary problem she can hire as many as she desires. A *selection criterion* is a mapping between test results of a single candidate to actions: $\texttt{Select}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau_i}) \in \{0, 1\}$, where 0 means *reject* and 1 means *accept* (hire). A *policy* π sets the selection criteria based on σ, p and other possible constraints such as probability to hire, error probability, etc. A *randomized threshold policy* is a policy π with parameters (τ, θ, r) such that $\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau_i}) = 1$ for $S_\tau := \sum_{i=1}^{\tau} \hat{y}_{i,j} > \theta$, $\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau_i}) = 0$ for $S_\tau < \theta$, and for $S_\tau = \theta$ the probability that $\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau_i}) = 1$ is r. We call a policy π a *threshold policy* if $r = 1$. In a *dynamic policy*, rather than setting a fixed number of tests per candidate, the employer may decide after each test whether to *accept*, *reject*, or to perform an additional test, i.e., $\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau_i}) \in \{0, 1, \texttt{more}\}$. Note that for a dynamic policy, the number of tests τ is a random variable determined based on the tests' outcomes. When designing a policy, one must carefully consider the balance between the following desiderata:

1. **Minimize False Discovery Rate (FDR)**—the fraction of unskilled workers among the accepted candidates, i.e., $\text{FDR} := \Pr[y_i = 0 | \pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = +1]$.
2. **Minimize False Omission Rate (FOR)**—the fraction of skilled workers among the rejected candidates, i.e., $\text{FOR} := \Pr[y_i = +1 | \pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 0]$.
3. **Minimize False Negatives (FN)**—the amount of skilled workers classified as unskilled.
4. **Minimize False Positives (FP)**—the amount of unskilled workers classified as skilled.

5. **Ratio of accept probability and number of tests**—the number of tests performed per candidate hired, using a parameter $B > 1$, we have $\frac{\tau}{B} \leq \Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = +1]$.

For any fixed number of tests τ, increasing the threshold θ of a threshold policy decreases FDR while increasing FOR.

Loss: To handle the trade-off between the false positives, (i.e., when an unskilled candidate is accepted) and false negatives (i.e., when a skilled candidate is rejected), we introduce an α-loss, paramaterized by $\alpha \in [0, 1]$ and defined as follows:

$$\ell_\alpha(b_1, b_2) = \alpha \cdot \mathbb{I}[b_1 = 0, b_2 = 1] + (1 - \alpha) \cdot \mathbb{I}[b_1 = 1, b_2 = 0]$$

where $\mathbb{I}[\cdot]$ is the indicator function and $b_1, b_2 \in \{0, 1\}$. The expected loss of a policy π is,

$$l_\alpha(\pi) = \mathbb{E}[\ell_\alpha(y_i, \pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}))] \tag{4.1}$$

where the expectation is over the type of the candidates y_i, the test results $\hat{y}_{i,j}$, and the decisions of π.

4.1.3 Analysis of the Bernoulli Model with One Group

To begin, we analyze this hiring model for a single group of candidates. The employer's goal is to minimize the expected loss, $l_\alpha(\pi)$, while maintaining a given acceptance probability.

The Simple Threshold Policy (Equal Number of Tests)

Consider the setting where the employer must subject all candidates to an equal number of tests τ and threshold θ (these parameters are chosen by the employer but thereafter constant across candidates). For a given threshold, we can relate the flip

probability (error rate) of the test to the probability that a candidate is accepted as follows:

Recall that $\hat{y}_{i,j} = y_i \oplus Br(\eta)$, $S_\tau = \sum_{j=1}^{\tau} \hat{y}_{i,j}$, that $Z_\tau^\eta = \sum_{t=1}^{\tau} \mathbb{I}(\hat{y}_{i,j} \neq y_i)$, and that τ and θ are the only parameters of the threshold policy, π. Informally, S_τ is the number of passed tests and Z_τ^η is the number of flips (tests in error). The probability of hiring an unskilled candidate is given by:

$$\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0] = \Pr[S_\tau \geq \theta | y_i = 0] = \Pr[Z_\tau^\eta \geq \theta].$$

Since Z_τ^η is a binomial random variable with parameters τ and η, we can calculate this probability precisely as:

$$\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0] = \Pr[Z_\tau^\eta \geq \theta] = \sum_{k=\theta}^{\tau} \binom{\tau}{k} \eta^k (1-\eta)^{\tau-k},$$

and the probability of rejecting a skilled candidate is the probability that they encounter more than $\tau - \theta$ flips, thus:

$$\begin{aligned} \Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 0 | y_i = +1] &= \Pr[S_\tau < \theta | y = +1] = \Pr[Z_\tau^\eta > \tau - \theta] \\ &= \sum_{k=\tau-\theta+1}^{\tau} \binom{\tau}{k} \eta^k (1-\eta)^{\tau-k}. \end{aligned}$$

Similarly, given a candidate's skill level, we can calculate the probability that they obtain exactly k positive tests out of τ, i.e,

$$\Pr[S_\tau = k | y_i = 0] = \Pr[Z_\tau^\eta = k] = \binom{\tau}{k} \eta^k (1-\eta)^{\tau-k}.$$

$$\Pr[S_\tau = k | y_i = +1] = \Pr[Z_\tau^\eta = \tau - k] = \binom{\tau}{k} \eta^{\tau-k} (1-\eta)^k.$$

Given these observations, we can now analyze the employer's choices.

Optimal solution for any ratio $\alpha \in (0,1)$

The next theorem shows that for threshold policies, the expected loss $l_\alpha(\pi) = l_\alpha(\theta)$ is minimized at $\theta^*_{p,\alpha}$ such that $|\theta^*_{p,\alpha} - \tau/2| \leq \frac{\log(\frac{1}{p})+\log(\frac{1}{\alpha})}{2\log(1+\frac{2\sigma}{1-\sigma})}$.

Theorem 4.1. *The loss function $l_\alpha(\theta)$ is quasi-convex and a threshold of*

$$\theta^*_{p,\alpha} = \arg\min_\theta l_\alpha(\theta) = \left\lceil \frac{\tau}{2} - \frac{\log(\frac{1}{p}-1)+\log(\frac{1}{\alpha}-1)}{2\log(1+\frac{2\sigma}{1-\sigma})} \right\rceil$$

minimizes loss for any values of $\alpha, p, \sigma \in (0,1)$.

Next, we bound the number of tests required to guarantee that the probability of classification error by the majority decision rule (i.e., $\theta = \lceil \frac{\tau}{2} \rceil$) does not exceed a specified quantity δ.

Theorem 4.2. *For every $\delta, p, \alpha \in (0,1)$, performing $\tau = \Omega(\frac{\alpha+p-2p\alpha}{\sigma^2}\ln(\frac{1}{\delta}))$ tests per candidate and using majority as a decision rule (i.e., $\theta = \tau/2$) guarantees $l_\alpha(\pi) \leq \delta$.*

Equal cost for false positives and false negatives $(\alpha = \frac{1}{2})$

Consider the simple loss consisting of the classification error rate (false positives and false negatives count equally), expressed via our loss function by setting $\alpha = \frac{1}{2}$. When skilled and unskilled candidates occur with equal frequency, i.e., $p = 1/2$, we can derive that the majority decision rule minimizes the classification error for any number of tests.

Corollary 4.3. *Assume $p = 1/2$ and $\alpha = 1/2$. For any number of tests τ, the majority decision rule minimizes loss l_α. Namely, $\arg\min_\theta l_{\frac{1}{2}}(\theta) = \lceil \frac{1}{2}\tau \rceil$. In addition, for every $\delta \in (0,1)$, performing $\tau = \Omega(\frac{1}{\sigma^2}\ln(\frac{1}{\delta}))$ tests per candidate and using majority*

as a decision rule guarantees classification error with probability of at most δ.

FDR minimization with limited number of tests per hire for balanced groups Again, assuming balanced groups (i.e., $p = 1/2$), suppose that an employer would like to minimize the false discovery rate, subject to the constraint of lower bounding the hiring probability. We can model this optimization problem by introducing a budget parameter $B > 1$ to bound any predetermined (fixed) number of tests per hired candidate as follows:

$$\begin{aligned} \arg\min_{\pi} \quad & \mathrm{FDR}_\pi = \Pr[y_i = 0 | \Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1] \\ \text{subject to} \quad & \frac{\tau_\pi}{\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1]} \leq B \end{aligned} \tag{4.2}$$

where τ_π is the number of tests π performs. The following theorem shows that the optimal policy is a randomized threshold policy.

Theorem 4.4. *There exists a randomized threshold policy π which is an optimal solution for (4.2).*

The Dynamic Policy (Adaptively-Allocated Tests)

Recall that under a dynamic policy, the employer can decide after each test whether to accept, reject, or perform another test. In general, dynamic policies are more efficient than those that must set a fixed number of tests. To build intuition, consider a candidate that has passed 2 out of 3 tests. As seen above, under an optimally-constructed fixed-test policy, any candidate that fails a single test might be rejected.[2] However, the posterior probability that this candidate is in fact *skilled* may still be greater than that of a fresh candidate sampled from the pool. Thus we can improve on the fixed-test policy by dynamically allocating more tests to candidates until their

[2]For example, if $B = 18$ and $\eta = \frac{1}{3}$, the lowest false discovery rate is achieved by $\tau = \theta = 3$.

posterior odds either dip below the prior odds or rise above the threshold for hiring. The following theorem formalizes this notion that it is better to administer more tests to a candidate that passed the majority of previous tests than to start afresh with a new candidate:

Theorem 4.5. *For any p, σ, τ, a candidate i that passed $\theta > \frac{\tau}{2}$ out of τ tests is more likely to be a skilled than a freshly-sampled candidate i' for whom no test results are yet available, i.e.,* $\Pr[y_{i'} = +1] = p < \Pr[y_i = +1 | S_\tau = \theta]$.

Remark 4.6. *If $\theta < \frac{\tau}{2}$, the inequality would have been reversed.*

The Greedy Policy We now present a greedy algorithm that continues to test a candidate so long as the posterior probability that $y_i = +1$ is greater than ϵ' and smaller than $1 - \epsilon$, rejects a candidate whenever the posterior falls below ϵ' (absent fairness concerns, employers will set $\epsilon' = p$ for all groups), and accepts whenever the posterior rises above $1 - \epsilon$. Given parameters $\epsilon, \epsilon' > 0$, we show that the greedy policy solves the optimization problem of minimizing the mean number of tests under these constraints, i.e.,

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & \mathbb{E}[\tau] \\ \text{subject to} \quad & \forall_i \pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 \text{ iff } \Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}] \geq 1 - \epsilon \\ & \forall_i \pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 0 \text{ iff } \Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}] < \epsilon' \end{aligned}$$

Our analysis of this policy builds upon the observation that conditioned on a worker's skill, the posterior log-odds after each test perform a one-dimensional random walk, starting with the prior log-odds $\log(\frac{p}{1-p})$ and moving, after each test result, either left (upon a failed test) or right (upon a passed test). When (as in our model) the probability of a flip are equal for skilled and unskilled candidates, our random walk has a fixed step size. Moreover, our random walk has *absorbing barriers* corresponding

to (when $\epsilon' = p$) falling below the prior log odds (on the left) and exceeding the hiring threshold (on the right). Owing to the fixed step size and absorbing barriers, our policy resembles the classic problem of Gambler's ruin, in which a gambler wins or loses a unit of currency at each step, and loses when crossing a threshold on the left (going bankrupt) or on the right (bankrupting the opponent). We formalize the random walk as follows where X_j is the position on the walk at time j:

1. X_0 is the prior log-odds of the candidate, i.e., $X_0 = \log \frac{p}{1-p}$.
2. After each test result, $\hat{y}_{i,j}$ is observed,

$$X_j = X_{j-1} + (2\hat{y}_{i,j} - 1) \cdot \log \left(\frac{\Pr[\hat{y}_{i,j} = +1 | y_i = +1]}{\Pr[\hat{y}_{i,j} = +1 | y_i = 0]} \right).$$

Let π_{Greedy} be the policy that accepts a candidate if $\Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,j}] \geq 1 - \epsilon$, rejects if $\Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,j}] < \epsilon'$, and otherwise conducts an additional test, i.e.,

$$\pi_{Greedy}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,j}) = \begin{cases} 0 & \text{if } \Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,j}] < \epsilon' \\ 1 & \text{if } \Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,j}] \geq 1 - \epsilon \cdot \\ \text{retest} & \text{else} \end{cases}$$

An employer will generally set the lower absorbing barrier to reject all candidates with posterior log odds less than p since a fresh candidate from the pool is expected to be better. However, when noise levels differ across groups, we may prefer *in the interest of fairness* to set ϵ' lower than p for members of the noisier group, allowing us to equalize the frequency of false negatives across groups (see Section 4.1.4).

Lemma 4.7. *Let $\beta, \beta' \in \mathbb{R}$ be the parameters that satisfy $\frac{\beta}{\beta+1} = 1 - \epsilon$ and $\frac{\beta'}{\beta'+1} = \epsilon'$ (i.e., $\beta = \frac{1-\epsilon}{\epsilon}$ and $\beta' = \frac{\epsilon'}{1-\epsilon'}$). Then $X_\tau \geq \log \beta$ iff $\Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}] \geq 1 - \epsilon$*

(iff the candidate is accepted) and $X_\tau < \log \beta'$ *iff* $\Pr[y_i = +1 | \hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}] < \epsilon'$ *(iff the candidate is rejected).*

Corollary 4.8. *The policy* π_{Greedy} *can be described as follows.*

$$\pi_{Greedy}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = \begin{cases} 0 & \text{if } X_\tau < \log \frac{\epsilon'}{1-\epsilon'} \\ 1 & \text{if } X_\tau \geq \log(\frac{1-\epsilon}{\epsilon}) \\ \text{retest} & \text{else} \end{cases}$$

We use the following parameters in the next theorems:

$$a = \left\lceil \frac{\log(\frac{(1-\epsilon)(1-\epsilon')(1+\sigma)}{\epsilon\epsilon'(1-\sigma)})}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil \gg \frac{1}{\sigma} \quad \text{and} \quad z = \left\lceil \frac{\log(\frac{p(1-\epsilon')(1+\sigma)}{\epsilon'(1-p)(1-\sigma)})}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil$$

Theorem 4.9 (Expected number of tests per type)**.** *The expected number of tests until a decision (namely accept or reject) for skilled candidates is*

$$\mathbb{E}[\tau_s] = \frac{1}{\sigma}\left(a \cdot \frac{1 - (\frac{1-\sigma}{1+\sigma})^z}{1 - (\frac{1-\sigma}{1+\sigma})^a} - z\right) \approx \frac{2a}{1+\sigma} - \frac{z}{\sigma},$$

and

$$\mathbb{E}[\tau_u] = \frac{1}{\sigma}\left(z - a \cdot \frac{1 - (\frac{1+\sigma}{1-\sigma})^z}{1 - (\frac{1+\sigma}{1-\sigma})^a}\right) \approx \frac{z}{\sigma}$$

for unskilled candidates.

For the probabilities of the candidates to be accepted or rejected, conditioned on their true skill level, we present the results in a form of confusion matrix in Table 4.1.

Theorem 4.10. *The expected number of tests until deciding whether to accept or reject a candidate is* $\mathbb{E}[\tau | \pi(y_{i,\tau}) \in \{0,1\}] \approx \frac{ap}{\sigma}$*, where* $a \gg \frac{1}{\sigma}$*.*

Table 4.1: Confusion matrix for π_{greedy} assuming $\epsilon \leq 1/4$ and $\epsilon' \leq p \leq 1/2$.

	General ϵ'		**When** $\epsilon' = p$	
	Skilled $(y_i = +1)$	**Unskilled** $(y_i = 0)$	**Skilled**	**Unskilled**
accept	$\text{TPR} = \Theta\left(1 - \frac{\epsilon'}{p}(1-\sigma)\right)$	$\text{FPR} = \Theta\left(\epsilon(p - \epsilon' + \epsilon'\sigma)\right)$	$\Theta(\sigma)$	$\Theta(\epsilon p \sigma)$
reject	$\text{FNR} = \Theta\left(\frac{\epsilon'}{p}(1-\sigma)\right)$	$\text{TNR} = \Theta\left(1 - \epsilon(p - \epsilon' + \epsilon'\sigma)\right)$	$\Theta(1-\sigma)$	$\Theta(1 - \epsilon p \sigma)$

4.1.4 Fairness Considerations in the Two-Group Setting

Two Groups—Threshold Policies We now discuss the effects of a threshold policy when candidates belong to two groups, G_1 and G_2 whose skill level is distributed identically, but whose tests are characterized by different noise levels. Without loss of generality, we assume that $\eta_1 < \eta_2$, where η_i is the probability that a test result of a candidate from G_i is different from his skill level. To begin, we note the fundamental irreconcilability of equalizing either the false positive or the false negative rates across groups with subjecting candidates to the same policy.

Theorem 4.11 (Impossibility result)**.** *When noise levels differ between two groups with identical skill level distribution, a single Threshold Policy π (with the same number of tests τ and the same threshold θ for both groups) cannot have equality in either the false negative rates or in the false positive rates across the groups. Particularly, there is a higher false positive rate in the noisier group, as an unskilled candidate from G_2 is more likely to be accepted by the threshold policy than an unskilled candidate from G_1:*

$$\text{FPR}^{\eta_1}_{\theta,\tau} = \Pr_{\eta_1}[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0] < \Pr_{\eta_2}[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0] = \text{FPR}^{\eta_2}_{\theta,\tau},$$

and also a higher false negative rate, as a skilled candidate from G_2 is more likely to

be rejected than a skilled candidate from G_1:

$$\begin{aligned}\mathrm{FNR}^{\eta_1}_{\theta,\tau} &= \Pr_{\eta_1}[\pi(\hat{y}_{i,1},\ldots,\hat{y}_{i,\tau})=0|y_i=+1] \\ &< \Pr_{\eta_2}[\pi(\hat{y}_{i,1},\ldots,\hat{y}_{i,\tau})=0|y_i=+1] = FNR^{\eta_2}_{\theta,\tau}.\end{aligned}$$

Connection to Economics Literature Aigner and Cain [2] discuss a similar case under a Gaussian screening model where the variance (noise level) of the single test differs across the two groups. Similarly, they note that qualified candidates fare worse in the noisy group but that unqualified candidates fare better in the noisier group. Our work differs from theirs in that we consider the effect of multiple tests and the ability to optimize over the number of tests.

Two Groups–Dynamic policy We now consider the (dynamic) hiring policy in the setting when employees belong to two groups, G_1 and G_2 with identically-distributed skills but different noise levels $\eta_1 < \eta_2$. We note that there are two ingredients that explain the differences among the groups: (i) The step size, $\log\left(\frac{\Pr[\hat{y}_{i,j}=+1|y_i=+1]}{\Pr[\hat{y}_{i,j}=+1|y_i=0]}\right) = \log\left(\frac{1-\eta}{\eta}\right)$ of G_2 (the noisier group) is smaller than the step size of G_1. Thus these candidates must typically pass more tests before they are accepted; and (ii) Skilled candidates in group G_2 exhibit less drift to the right (they have a higher probability of failing a test). Consequently, when an employer (rationally) sets $\epsilon' = p$ for all groups, a skilled candidate from G_2 is more likely to be fail a test in step 1, at which point the dynamic policy summarily rejects them. These two facts explain both the higher false negative rates for G_2 and the longer expected duration until acceptance. By setting $\epsilon' < p$ for members of the noisier group, we can equalize false negative rates. Precisely, setting $\epsilon' = \frac{\eta_1}{\eta_2}p$ achieves the desired parity. The cost of this intervention is that it requires more tests for candidates from the noisier group. Here, our random walk analysis can be leveraged to determine exactly how many more.

Once again, we cannot provide equality across the groups in all desired ways—the same acceptance criterion, the same expected number of tests, and the same false negative rates between groups—with the noise differs across groups.

4.1.5 Candidate Screening: Gaussian Model

In this section, we work out the analytic solutions for the conditional expectation of worker qualities given a series of conditionally independent tests $Y_1, ...Y_n$ s.t. $\forall i, j$, $Y_i \perp Y_j | Q$. We assume that the worker quality Q normally distributed with mean μ_Q and variance σ_Q^2, so instead of binary skill level we have continuous quality of candidates. Conditioned on $Q = q$, each test is generated according to the structural equation $y_i = q + \eta$, where η is a normally distributed noise term with mean 0 and variance σ_η^2. Equivalently, we can say that the conditional distribution for each test $P(Y|Q = q)$ is Gaussian with mean q and variance σ_η^2. We refer the reader to the full version [38] for further details.

We show that we can equalize conditional variance between the two groups by giving more interviews to noisier group, and that it yields the same conditional expectations.

Theorem 4.12. *For two groups, G_1, G_2 with the same worker quality Q, that differ only in the variance of their noise $\sigma_{\eta_1}^2 < \sigma_{\eta_2}^2$, the variance can be equalized by using $n_2 = \frac{\sigma_{\eta_2}^2}{\sigma_{\eta_1}^2} n_1$ interviews (or tests) for G_2, where n_1 is the number of interviews for each candidate from G_1.*

Theorem 4.13. *When equalizing conditional variances between G_1, G_2 by using $n_2 = \frac{\sigma_{\eta_2}^2}{\sigma_{\eta_1}^2} n_1$, we get the same conditional expectations, $\mathbb{E}_{\eta_1}[Q|Y_1, ..., Y_{n_1}] = \mathbb{E}_{\eta_2}[Q|Y_1, ..., Y_{n_2}]$.*

4.1.6 Unsupervised Parameter Estimation

Now, under the assumption of realizable case, we explain how one can estimate the parameters p and σ given tests results from a homogeneous population. Surprisingly, we discover that parameter recovery in this model does not require any ground truth labels indicating whether an employee is skilled or unskilled. We use Hoeffding's inequality to bound the absolute difference between the estimated parameters and the true parameters by choosing δ as the wanted upper bound and solving for the number of samples or ϵ.

Lemma 4.14 (Hoeffding's inequality)**.** *Let $y_1, \ldots, y_m$ be σ^2-sub–gaussian random variables. Then, for any $\epsilon > 0$,*

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} y_i - \mathbb{E}[y_i]\right| \geq \epsilon\right] \leq 2e^{-m\epsilon^2/2\sigma^2}.$$

If $y_1, \ldots, y_m$ are Bernoulli random variables with parameter p,

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} y_i - p\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}.$$

We start by estimating σ and then use it to derive an estimate for p. The estimated parameters are denoted by $\hat{\sigma}$ and $\hat{p}$. Notice that in order to have any information regarding the true value of σ, we need to have candidates with at least two tests. Hence, from now on we assume exactly that, i.e., $\forall_i \pi_{\text{Greedy}}(\hat{y}_{i,1}) = more$ for dynamic policies and $\tau \geq 2$ for fixed number of tests policies.

Now, in both policies we have showed that the optimal rule is to reject candidates that fail their first test. Therefore inconsistencies between the first two tests are seen only in cases where $\hat{y}_{i,1} = 1, \hat{y}_{i,2} = 0$.

Let c be the number of inconsistencies in the first two tests, i.e., $c = |\{(\hat{y}_{i,1}, \hat{y}_{i,2}) :$

$y_{i,1} \neq y_{i,2}\}|$, and let m be the number of candidates with at least two tests. Since c is generated by sampling m times, the distribution $Br((\frac{1+\sigma}{2})(\frac{1-\sigma}{2})) = Br(\frac{1-\sigma^2}{4})$ and we can estimate σ as stated in the next theorem:

Theorem 4.15. *If we have results from $m \geq \frac{1}{2\epsilon^2}\ln\frac{2}{\delta}$ candidates, by using $\hat{\sigma} = \sqrt{1-4\frac{c}{m}}$, then with probability $1-\delta$ we have that $|\hat{\sigma}-\sigma| \leq \epsilon$.*

Having an estimation of the parameter $\hat{\sigma}$, we can calculate the estimated p as follows: Let $p_{\hat{y}_{*,1}=1} := \frac{\sum_i \mathbb{I}(\hat{y}_{i,1}=1)}{m}$ be the percentage of positive first tests. Since this number is generated by the distribution $Br(\frac{1}{2}(p(1+\sigma)+(1-p)(1-\sigma))) = Br(\frac{1}{2}+(2p-1)\frac{\sigma}{2})$, we can estimate $\hat{p}$ using the estimated value of $\hat{\sigma}$.

Theorem 4.16. *If we have results from $m \geq \frac{1}{2\epsilon^2}\ln\frac{2}{\delta}$ candidates, by using $\hat{p} = \frac{2(p_{y_{*,1}=1}-1)+\hat{\sigma}}{\hat{\sigma}}$, we get that with probability $1-\delta$ we have that $|\hat{p}-p| \leq 2\epsilon$.*

Under the Gaussian screening model, the parameter estimation is also straightforward (assuming realizability) without access to the true skill level of the employees. We start by looking at a single candidate, i. Each of his test results, $\hat{y}_{i,j}$ is generated from a conditional distribution $P(Y_i|Q_i = q_i)$ which is a Gaussian with mean q_i and variance σ_η^2. Since this variance is common among all the candidates, we can simply average the estimated variance of every candidate to get an approximation for σ_η^2. Suppose $\hat{y}_{i,1},\ldots,\hat{y}_{i,n}$ is a sequence of n i.i.d tests of candidate i, and let $\boldsymbol{y_i} = \frac{1}{n}\sum_{j=1}^{n} y_{i,j}$ be the empirical mean of candidate i's tests.

The following theorem is a result from Hoeffding's Inequality, in which we use to bound the error of our estimated parameters.

Theorem 4.17. *By using the following as estimators for Gaussian parameters $\hat{\mu}_Q = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y_i}$, $\hat{\sigma}_\eta^2 = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}(y_{i,j}-\boldsymbol{y_i})^2$ and $\hat{\sigma}_Q^2 = \frac{1}{m}\sum_{i=1}^{m}(\hat{\mu}_Q-\boldsymbol{y_i})^2$ (notice that $\mathbb{E}[\hat{\sigma}_\eta^2] = \sigma_\eta^2$ and $\mathbb{E}[\hat{\sigma}_Q^2] = \sigma_Q^2$), the difference between each parameter and it's estimator is bounded by $O(\sqrt{\frac{1}{m}\ln(\frac{1}{\delta})})$.*

4.1.7 Proofs for One Group Setting (Section 4.1.3)

Proof of Theorem 4.1. To prove the theorem, we show that the loss function $l_\alpha(\tau, \theta)$, as a function of θ is quasi-convex and achieves its minimum value at

$$\left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p} - 1) + \log(\frac{1}{\alpha} - 1)}{\log(1 + \frac{2\sigma}{1-\sigma})}) \right\rceil .$$

Namely, we show that the loss is monotone increasing for

$$\left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p} - 1) + \log(\frac{1}{\alpha} - 1)}{\log(1 + \frac{2\sigma}{1-\sigma})}) \right\rceil \leq \theta \leq \tau - 1,$$

i.e., increasing θ increases the loss: $l_\alpha(\theta) < l_\alpha(\theta + 1)$.
Similarly, we show that for

$$1 \leq \theta \leq \left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p} - 1) + \log(\frac{1}{\alpha} - 1)}{\log(1 + \frac{2\sigma}{1-\sigma})}) \right\rceil ,$$

we have $l_\alpha(\theta) < l_\alpha(\theta - 1)$.
Indeed,

$$l_\alpha(\theta + 1, \tau) - l_\alpha(\theta, \tau) = -\alpha \Pr[y = 0, S_\tau = \theta] + (1 - \alpha) \Pr[y = +1, S_\tau = \theta]$$

$$= -\alpha \Pr[S_\tau = \theta | y = 0] \Pr[y = 0] + (1 - \alpha) \Pr[S_\tau = \theta | y = +1] \Pr[y = +1]$$

Since $\Pr[y = 0] = 1 - p$ and $\Pr[y = +1] = p$, we have

$$l_{\frac{1}{2}}(\theta + 1, \tau) - l_{\frac{1}{2}}(\theta, \tau) = -(1 - p)\alpha \Pr[S_\tau = \theta | y = 0] + p(1 - \alpha) \Pr[S_\tau = \theta | y = +1].$$

The above expression is positive iff

$$(1-p)\alpha \Pr[S_\tau = \theta | y = 0] < p(1-\alpha) \Pr[S_\tau = \theta | y = +1] \tag{4.3}$$

Since $\Pr[S_\tau = \theta | y = 0]$ is the probability of exactly θ flips, and $\Pr[S_\tau = \theta | y = +1]$ is the probability of exactly $\tau - \theta$ flips, we can calculate those probabilities as follows:

$$\Pr[S_\tau = \theta | y = 0] = \binom{\tau}{\theta} (\frac{1-\sigma}{2})^\theta (\frac{1+\sigma}{2})^{\tau-\theta}$$

$$\Pr[S_\tau = \theta | y = +1] = \binom{\tau}{\tau-\theta} (\frac{1-\sigma}{2})^{\tau-\theta} (\frac{1+\sigma}{2})^\theta$$

Substituting expression in (4.3), we get

$$(1-p)\alpha \binom{\tau}{\theta} (\frac{1-\sigma}{2})^\theta (\frac{1+\sigma}{2})^{\tau-\theta} < p(1-\alpha) \binom{\tau}{\tau-\theta} (\frac{1-\sigma}{2})^{\tau-\theta} (\frac{1+\sigma}{2})^\theta.$$

Rearranging, we get

$$(\frac{1-\sigma}{1+\sigma})^{2\theta} < (\frac{1-\sigma}{1+\sigma})^\tau (\frac{p}{1-p})(\frac{1-\alpha}{\alpha}).$$

Applying log on both sides gets us

$$2\theta \log(\frac{1-\sigma}{1+\sigma}) < \tau \log(\frac{1-\sigma}{1+\sigma}) + \log(\frac{p}{1-p}) + \log(\frac{1-\alpha}{\alpha}).$$

Solving for θ, we find that the inequality holds if

$$\theta > \frac{\tau \log(\frac{1-\sigma}{1+\sigma}) + \log(\frac{p}{1-p}) + \log(\frac{1-\alpha}{\alpha})}{2\log(\frac{1-\sigma}{1+\sigma})} = \left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p}-1) + \log(\frac{1}{\alpha}-1)}{\log(1+\frac{2\sigma}{1-\sigma})}) \right\rceil$$

For $\theta \geq \left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p}-1)+log(\frac{1}{\alpha}-1)}{\log(1+\frac{2\sigma}{1-\sigma})}) \right\rceil$, we have

$$(1-p)\alpha \Pr[S_\tau = \theta | y = 0] < p(1-\alpha) \Pr[S_\tau = \theta | y = +1],$$

and for $\theta \leq \left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p}-1)+\log(\frac{1}{\alpha}-1)}{\log(1+\frac{2\sigma}{1-\sigma})}) \right\rceil$, we have

$$\alpha(1-p) \Pr[S_\tau = \theta | y = 0] > (1-\alpha)p \Pr[S_\tau = \theta | y = +1].$$

This implies that the maximum is $\theta^*_{p,\alpha} = \left\lceil \frac{1}{2}(\tau - \frac{\log(\frac{1}{p}-1)+\log(\frac{1}{\alpha}-1)}{\log(1+\frac{2\sigma}{1-\sigma})}) \right\rceil$.

■

Proof of Theorem 4.2. We start with a skilled candidate. The expected number of tests that a skilled candidate passes is $\mathbb{E}[S_\tau | y = +1] = \tau(\frac{1+\sigma}{2}) > \frac{\tau}{2}$.

By using Hoeffding's inequality for Bernoulli distributions, for every $\epsilon > 0$,

$$\Pr[\mathbb{E}[S_\tau] - S_\tau \geq \epsilon | y = +1] = \Pr[\tau(\frac{1+\sigma}{2}) - S_\tau \geq \epsilon | y = +1] \leq e^{-2\epsilon^2 \tau} < \delta.$$

Choosing $\epsilon = \frac{\sigma}{2}$ yields $S_\tau \leq \frac{\tau}{2} < \lceil \frac{\tau}{2} \rceil$ (as τ is odd), which holds iff a majority threshold policy would predict that this is an unskilled candidate (false negative). Solving for τ, we get $\tau > \frac{1}{\sigma^2} \ln(\frac{1}{\delta})$.

We now repeat the process for an unskilled candidate. The expected number of tests that an unskilled candidate passes is $\mathbb{E}[S_\tau | y = 0] = \tau(\frac{1-\sigma}{2}) < \frac{\tau}{2}$.

By using Hoeffding's inequality again, we have

$$\Pr[S_\tau - \mathbb{E}[S_\tau] \geq \epsilon | y = 0] = \Pr[S_\tau - \tau(\frac{1-\sigma}{2}) \geq \epsilon | y = 0] \leq e^{-2\epsilon^2 \tau} < \delta$$

Choosing $\epsilon = \frac{\sigma}{2}$ yields $S_\tau > \frac{\tau}{2}$, which holds iff a majority threshold falsely predicts that

this is a skilled candidate (false positive). Solving for τ again, we get $\tau > \frac{1}{\sigma^2}\ln(\frac{1}{\delta})$. Overall, $\tau > \frac{\alpha(1-p)}{\sigma^2}\ln(\frac{1}{\delta}) + \frac{p(1-\alpha)}{\sigma^2}\ln(\frac{1}{\delta}) = \Omega(\frac{\alpha+p-2p\alpha}{\sigma^2}\ln(\frac{1}{\delta}))$ ∎

Proof of Theorem 4.4. Let π' be any optimal policy for (4.2) (not necessarily threshold) with a fixed number of tests, τ. We will show, in two steps, how to transform it into an optimal randomized threshold policy. The first step is to symmetrize π'. Let $r_k = \Pr[\pi(\hat{y}) = 1 | S_\tau = k]$. Define a policy π'', which performs τ tests, and accepts with probability r_k where $k = S_\tau$. Clearly, both π' and π'' have the same accept probability. In addition, since condition on $S_\tau = k$, any sequence of outcomes is equally likely. Furthermore, and the probability that $y = 1$ given any sequence of outcomes with $S_\tau = k$, is identical. (Technically, S_τ is a sufficient statistics.) This implies that the false discovery rate is also unchanged.

This yields that π with the randomization vector r is also optimal.

The second step is to suppose—for sake of contradiction—that π'' is not a randomized threshold policy. We will show that we can improve the FDR of π'' while keeping the probability of acceptance unchanged. This will contradict the hypothesis that π' is optimal.

If π'' is not a randomized threshold policy, then there is no θ and k, such that

$$r_k = \Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | S_\tau = k \neq \theta] = \begin{cases} 0, & \text{if } k < \theta \\ 1 & \text{if } k > \theta \end{cases}.$$

Now, let k be the minimal value such that $r_k > 0$ and let $0 < i < \tau - k$ be the minimal value for which $0 < r_{k+i} < 1$. Clearly, the FDR is lower at $S_\tau = k + i$ than at $S_\tau = k$. Intuitively, we can shift some probability mass, $\epsilon_k > 0$ from r_k to r_{k+i} in a way that maintains the acceptance probability of π and decreases the false positive

rates.

Let $\epsilon_{k+i} > 0$ be such that $\epsilon_k \cdot r_k = \epsilon_{k+i} \cdot r_{k+i}$. Let r' be a modified randomization vector for π such that $r'_k = r_k(1-\epsilon_k), r'_{k+i} = r_{k+i}(1+\epsilon_{k+i})$ and for every $l \notin \{k, k+i\}$ $r'_l = r_l$. Since $\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1] = \sum_{l=1}^{\tau} r_l = \sum_{l \notin \{k,k+i\}} r_l + r'_k + r'_{k+i}$, the acceptance probability remains the same. As for the false discovery rate, since $\Pr[y_i = 0 | S_\tau = k+i] < \Pr[y_i = 0 | S_\tau = k]$, $\Pr[S_\tau = k+i]$ is higher with r' than with r, $\Pr[S_\tau = k]$ is lower with r' than with r and for any $l \notin \{k, k+i\}$, $\Pr[S_\tau = l]$ with r' is the same as with r, the false discovery rate with r' is lower, which contradicts the optimality of π with r as the randomization vector. ■

Proof of Theorem 4.5. Using Bayes' theorem, the conditional probability can be decomposed as

$$\Pr[y_i = +1 | S_\tau = \theta] = \frac{\Pr[y_i = +1] \Pr[S_\tau = \theta | y_i = +1]}{\Pr[S_\tau = \theta]} =$$

$$\frac{p\binom{\tau}{\theta}(\frac{1-\sigma}{2})^{\tau-\theta}(\frac{1+\sigma}{2})^{\theta}}{p\binom{\tau}{\theta}(\frac{1-\sigma}{2})^{\tau-\theta}(\frac{1+\sigma}{2})^{\theta} + (1-p)\binom{\tau}{\tau-\theta}(\frac{1+\sigma}{2})^{\tau-\theta}(\frac{1-\sigma}{2})^{\theta}}.$$

Since $\tau - \theta < \theta$ and $\binom{\tau}{\theta} = \binom{\tau}{\tau-\theta}$, we get

$$\frac{p(1+\sigma)^{2\theta-\tau}}{p(1+\sigma)^{2\theta-\tau} + (1-p)(1-\sigma)^{2\theta-\tau}} = \frac{p(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau}}{p(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau} + 1 - p}.$$

Since $(\frac{1+\sigma}{1-\sigma}) > 1$ it holds that $(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau} > 1$,

$$(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau}(1-p) > 1-p.$$

So,

$$(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau} > p(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau} + 1 - p,$$

And finally,

$$\Pr[y_{i'} = +1] = p < \frac{p(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau}}{p(\frac{1+\sigma}{1-\sigma})^{2\theta-\tau} + 1 - p} = \Pr[y_i = +1 | S_\tau = \theta].$$

■

Proof of Lemma 4.7. Let $S'_\tau = \sum_{j=1}^{\tau}(2\hat{y}_{i,j} - 1)$, and let $s_\tau \in \{-\tau, \ldots, \tau\}$ be any of the possible values of S'_τ. Note that

$$\frac{\Pr[\hat{y}_{i,j} = 1 | y_i = 1]}{\Pr[\hat{y}_{i,j} = 1 | y_i = 0]} = \frac{1+\sigma}{1-\sigma}.$$

Since the $\hat{y}_{i,j}$ are i.i.d., we have

$$\begin{aligned} X_\tau =& X_0 + \sum_{j=1}^{\tau}(2\hat{y}_{i,j} - 1) \cdot \log(\frac{\Pr[\hat{y}_{i,j} = +1 | y_i = +1]}{\Pr[\hat{y}_{i,j} = +1 | y_i = 0]}) \\ =& \log(\frac{p}{1-p}) + S_\tau \log(\frac{1+\sigma}{1-\sigma}) \\ =& \log((\frac{p}{1-p})(\frac{1+\sigma}{1-\sigma})^{S_\tau}). \end{aligned}$$

Since

$$\frac{\Pr[S_\tau = s_\tau | y_i = 1]}{\Pr[S_\tau = s_\tau | y_i = 0]} = (\frac{1+\sigma}{1-\sigma})^{s_\tau},$$

we have

$$X_\tau = \log((\frac{p}{1-p})(\frac{\Pr[S_\tau = s_\tau | y_i = 1]}{\Pr[S_\tau = s_\tau | y_i = 0]})). \tag{4.4}$$

Since

$$\Pr[S_\tau = s_\tau | y_i = 1] = \frac{\Pr[S_\tau = s_\tau] \cdot \Pr[y_i = 1 | S_\tau = s_\tau]}{\Pr[y_i = 1]}$$

and

$$\Pr[S_\tau = s_\tau | y_i = 0] = \frac{\Pr[S_\tau = s_\tau] \cdot \Pr[y_i = 0 | S_\tau = s_\tau]}{\Pr[y_i = 0]},$$

assigning $\Pr[y_i = 0] = 1 - p$ and $\Pr[y_i = 1] = p$, we get

$$\frac{\Pr[S_\tau = s_\tau | y_i = 1]}{\Pr[S_\tau = s_\tau | y_i = 0]} = \frac{(1-p) \cdot \Pr[y_i = 1 | S_\tau = s_\tau]}{p \cdot \Pr[y_i = 0 | S_\tau = s_\tau]}. \tag{4.5}$$

Applying (4.5) in (4.4) and adding $X_\tau \geq \log \beta$ gives us

$$X_\tau = \log \left(\frac{\Pr[y_i = 1 | S_\tau = s_\tau]}{\Pr[y_i = 0 | S_\tau = s_\tau]} \right) = \log \left(\frac{\Pr[y_i = 1 | S_\tau = s_\tau]}{1 - \Pr[y_i = 1 | S_\tau = s_\tau]} \right) \geq \log \beta$$

$$\frac{\Pr[y_i = 1 | S_\tau = s_\tau]}{1 - \Pr[y_i = 1 | S_\tau = s_\tau]} \geq \beta$$

$$\Pr[y_i = 1 | S_\tau = s_\tau] \geq \beta(1 - \Pr[y_i = 1 | S_\tau = s_\tau])$$

$$\Pr[y_i = 1 | S_\tau = s_\tau] \geq \frac{\beta}{1+\beta}$$

Applying (4.5) in (4.4) and adding $X_\tau < \log \beta'$ gives us

$$\frac{\Pr[y_i = 1 | S_\tau = s_\tau]}{1 - \Pr[y_i = 1 | S_\tau = s_\tau]} < \beta'$$

Hence

$$\Pr[y_i = 1 | S_\tau = s_\tau] < \frac{\beta'}{1+\beta'}$$

■

Proof of Theorem 4.9. First recall that given a skilled candidate, for every test j,

$$\Pr[\hat{y}_{i,j} = +1 | y_i = +1] = \frac{1+\sigma}{2}$$

$$\Pr[\hat{y}_{i,j} = 0 | y_i = +1] = \frac{1-\sigma}{2}$$

Hence

$$\Pr[\hat{y}_{i,j} = 0 | y_i = 1] - \Pr[\hat{y}_{i,j} = +1 | y_i = 1] = -\sigma.$$

The lower absorbing barrier is reached when a candidate's posterior skill level is lower than the prior of the skill level, i.e.,

$$\log \frac{\epsilon'}{1-\epsilon'} - \log\left(\frac{1+\sigma}{1-\sigma}\right)$$

and the starting point is just one step away from the lower absorbing barrier:

$$X_0 = \log \frac{p}{1-p}.$$

According to Corollary 4.8, the upper absorbing barrier is in

$$\log(\frac{1-\epsilon}{\epsilon}).$$

To derive the results for the expected duration of the random walk for skilled and unskilled candidates, we shift the locations of the absorbing points so that the lower barrier would be in 0 and also divide them by a step size (so now we have that every step is of size 1). The new upper absorbing barrier is at

$$a = \left\lceil \frac{\log(\frac{1-\epsilon}{\epsilon}) - (\log \frac{\epsilon'}{1-\epsilon'} - \log(\frac{1+\sigma}{1-\sigma}))}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil = \left\lceil \frac{\log(\frac{(1-\epsilon)(1-\epsilon')(1+\sigma)}{\epsilon\epsilon'(1-\sigma)})}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil.$$

And we also shift the starting point:

$$z = \left\lceil \frac{\log \frac{p}{1-p} - (\log \frac{\epsilon'}{1-\epsilon'} - \log(\frac{1+\sigma}{1-\sigma}))}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil = \left\lceil \frac{\log(\frac{p(1-\epsilon')(1+\sigma)}{\epsilon'(1-p)(1-\sigma)})}{\log(\frac{1+\sigma}{1-\sigma})} \right\rceil$$

As stated in [55], the expected duration of a random walk with absorbing barriers of

0 and a from $z = 1$ is (equation 3.4, chapter XIV [page 348]):

$$\mathbb{E}[\tau_s] = \mathbb{E}[D_{z=1}] = \frac{1}{q-p}\left(z - a \cdot \frac{1-(\frac{q}{p})^z}{1-(\frac{q}{p})^a}\right) = \frac{1}{-\sigma}\left(z - a \cdot \frac{1-(\frac{1-\sigma}{1+\sigma})^z}{1-(\frac{1-\sigma}{1+\sigma})^a}\right).$$

Hence,

$$\mathbb{E}[\tau_s] = \frac{1}{\sigma}\left(a \cdot \frac{1-(\frac{1-\sigma}{1+\sigma})^z}{1-(\frac{1-\sigma}{1+\sigma})^a} - z\right).$$

As for unskilled candidates, the absorbing points and the starting point are the same, the only difference is that

$$\Pr[\hat{y}_{i,j} = +1 | y_i] = \frac{1-\sigma}{2}$$

and

$$\Pr[\hat{y}_{i,j} = 0 | y_i = +1] = \frac{1+\sigma}{2}.$$

Therefore,

$$\Pr[\hat{y}_{i,j} = 0 | y_i = 0] - \Pr[\hat{y}_{i,j} = +1 | y_i = 0] = \sigma$$

and we deduce

$$\mathbb{E}[\tau_u] = \frac{1}{\sigma}\left(z - a \cdot \frac{1-(\frac{1+\sigma}{1-\sigma})^z}{1-(\frac{1+\sigma}{1-\sigma})^a}\right).$$

■

Deviations for the confusion matrix (Table 4.1). We split the claim in the confusion matrix (Table 4.1) into two parts. First, using equation (2.4) from chapter XIV [page 345] in [55], we get

$$\text{FNR} = \Pr[\pi_{\text{Greedy}}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 0 | y_i = +1] = \frac{(\frac{1-\sigma}{1+\sigma})^a - (\frac{1-\sigma}{1+\sigma})^z}{(\frac{1-\sigma}{1+\sigma})^a - 1}$$

and

$$\mathrm{TNR} = \Pr[\pi_{\mathrm{Greedy}}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 0 | y_i = 0] = \frac{(\frac{1+\sigma}{1-\sigma})^a - (\frac{1+\sigma}{1-\sigma})^z}{(\frac{1+\sigma}{1-\sigma})^a - 1}.$$

The second part follows from the fact the gambler's ruin must end in case of absorbing barriers.

$$\mathrm{TPR} = \Pr[\pi_{\mathrm{Greedy}}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = +1] = 1 - \frac{(\frac{1-\sigma}{1+\sigma})^a - (\frac{1-\sigma}{1+\sigma})^z}{(\frac{1-\sigma}{1+\sigma})^a - 1} =$$

$$\frac{(\frac{1-\sigma}{1+\sigma})^z - 1}{(\frac{1-\sigma}{1+\sigma})^a - 1} = \frac{\frac{\epsilon'(1-p)(1-\sigma)}{p(1-\epsilon')(1+\sigma)} - 1}{\frac{\epsilon'\epsilon(1-\sigma)}{(1-\epsilon')(1-\epsilon)(1+\sigma)} - 1} = \frac{\frac{\mu(1-p)}{p} - 1}{\frac{\epsilon\mu}{(1-\epsilon)} - 1} = \frac{(1-\epsilon)(\mu(1-p) - p)}{p(\epsilon\mu - (1-\epsilon))},$$

Where $\mu := \frac{\epsilon'(1-\sigma)}{(1-\epsilon')(1+\sigma)}$. For $\epsilon \leq 1/4$ and $p < 1/2$ we get $0 \leq \mu \leq 1/3$ and $\mu = \Theta(\epsilon'(1-\sigma))$, therefore

$$\mathrm{TPR} = \Theta\left(\frac{p-\mu}{p}\right) = \Theta\left(1 - \frac{\epsilon'}{p}(1-\sigma)\right).$$

Hence $\mathrm{FNR} = \Theta(\frac{\epsilon'}{p}(1-\sigma))$.

$$\mathrm{FPR} = \Pr[\pi_{\mathrm{Greedy}}(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0] = \frac{(\frac{1+\sigma}{1-\sigma})^z - 1}{(\frac{1+\sigma}{1-\sigma})^a - 1} = \frac{\frac{p(1-\epsilon')(1+\sigma)}{(1-p)\epsilon'(1-\sigma)} - 1}{\frac{(1-\epsilon')(1-\epsilon)(1+\sigma)}{\epsilon'\epsilon(1-\sigma)} - 1} =$$

$$= \frac{\frac{p}{(1-p)\mu} - 1}{\frac{(1-\epsilon)}{\epsilon\mu} - 1} \frac{\epsilon(p - (1-p)\mu)}{(1-p)(1-\epsilon-\epsilon\mu)} = \Theta\left(\epsilon(p-\mu)\right) = \Theta\left(\epsilon(p - \epsilon' + \epsilon'\sigma)\right)$$

Hence $\mathrm{TNR} = \Theta\left(1 - \epsilon(p - \epsilon' + \epsilon'\sigma)\right)$. ∎

Proof of Theorem 4.10.

$$\mathbb{E}[\tau] = \mathbb{E}[\tau_s]p + \mathbb{E}[\tau_u](1-p) =$$

$$= \frac{1}{\sigma}\left(a \cdot \frac{1-(\frac{1-\sigma}{1+\sigma})^z}{1-(\frac{1-\sigma}{1+\sigma})^a} - z\right)p + \frac{1}{\sigma}\left(z - a \cdot \frac{1-(\frac{1+\sigma}{1-\sigma})^z}{1-(\frac{1+\sigma}{1-\sigma})^a}\right)(1-p) =$$

$$\approx \frac{1}{\sigma}\left(a \cdot (1 - \frac{\epsilon'}{p}(1-\sigma)) - z\right)p + \frac{1}{\sigma}(z - a(\epsilon(p - \epsilon' + \epsilon'\sigma)))(1-p) \approx \frac{ap}{\sigma}$$

■

4.1.8 Proofs for Two Groups Setting (Section 4.1.4)

The next lemma aids in the proof of Theorem 4.11.

Lemma 4.18. *Let Z_n^η be a Binomial random variable with parameters $n \in \mathbb{N}$ and $\eta \in (0,1)$. Given a number of successes, $k \in \{0,\ldots,n\}$, we know that the probability mass function of Z_n^η is $f_k(\eta) := \Pr[Z_n^\eta = k] = \binom{n}{k}\eta^k(1-\eta)^{n-k}$. Let $\mathcal{L}(\eta|k)$ be the likelihood function of the event $Z_n^\eta = k$. Then the maximum likelihood of $f_k(\eta)$ is $\eta = \frac{k}{n}$. I.e.,*

$$\mathcal{L}(\eta|k) = argmax_\eta f_k(\eta) = argmax_\eta \binom{n}{k}\eta^k(1-\eta)^{n-k} = \frac{k}{n}.$$

Proof of Lemma 4.18. We notice that $\binom{n}{k}$ does not depend on η, thus

$$\text{argmax}_\eta f_k(\eta) = \text{argmax}_\eta \binom{n}{k}\eta^k(1-\eta)^{n-k} = \text{argmax}_\eta \eta^k(1-\eta)^{n-k}$$

The log-likelihood is particularly convenient for maximum likelihood estimation. Logarithms are strictly increasing functions, as a result, maximizing the likelihood is equivalent to maximizing the log-likelihood, i.e.,

$$\text{argmax}_\eta \eta^k(1-\eta)^{n-k} = \text{argmax}_\eta \ln(\eta^k(1-\eta)^{n-k}) = \text{argmax}_\eta k\ln(\eta) + (n-k)\ln(1-\eta)$$

Differentiating (with respect to η) and comparing to zero we get

$$\frac{d\ln(f_k(\eta))}{d\eta} = \frac{k}{\eta} - \frac{n-k}{1-\eta} = 0.$$

And after refactoring,

$$k(1-\eta) = (n-k)\eta$$

The function $\ln(f_k(\eta))$ is a strictly concave as its second derivative is negative,

$$\frac{d^2\ln(f_k(\eta))}{d\eta^2} = -\frac{k}{\eta^2} - \frac{n-k}{(1-\eta)^2} < 0,$$

And since the derivative of a strictly concave function is zero at $\frac{k}{n}$, then $\hat{\eta} = \frac{k}{n}$ is a global maximum. Therefore, $\hat{\eta} = \frac{k}{n}$ obtains absolute maximum in $f_k(\eta)$. ■

Proof of Theorem 4.11. Let $Z_\tau^{\eta_i}$ be a random variable that represents the number of flips out of a τ-tests sequence with a noise level of η_i, i.e., $Z_\tau^{\eta_i}$ is the number of times when $y_j \neq y$ for $1 \leq j \leq \tau$. We use $Z_\tau^{\eta_i}$ to express $\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0, \eta = \eta_i]$ as the probability that at least θ flips,

$$\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = 0, \eta = \eta_i] = \Pr[Z_\tau^{\eta_i} \geq \theta]$$

and the probability of $\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | q = +1, \eta = \eta_i]$ as at most $\tau - \theta$ flips, thus

$$\Pr[\pi(\hat{y}_{i,1}, \ldots, \hat{y}_{i,\tau}) = 1 | y_i = +1, \eta = \eta_i] = \Pr[Z_\tau^{\eta_i} \leq \tau - \theta].$$

From Lemma (4.18) and since probability density function (pdf) are is monotone increasing, we derive that the pdf of $Z_n^{\eta_2}$ satisfies *monotone likelihood ratio property* over the pdf of $Z_n^{\eta_1}$. This implies that the pdf of $Z_n^{\eta_2}$ also has *first-order stochastic dominance* over $Z_n^{\eta_1}$ by Theorem 1.1 in [128]. From *stochastic dominance*, we can

derive the desired inequalities

$$FP_{\theta,\tau}^{\eta_1} = \Pr[\theta \leq Z_n^{\eta_1}] < \Pr[\theta \leq Z_n^{\eta_2}] = FP_{\theta,\tau}^{\eta_2}$$

and

$$FN_{\theta,\tau}^{\eta_1} = \Pr[Z_n^{\eta_1} \leq \tau - \theta] < \Pr[Z_n^{\eta_2} \leq \tau - \theta] = FN_{\theta,\tau}^{\eta_2}.$$

■

4.1.9 Proofs for the Gaussian Setting (Section 4.1.5)

Proof of Theorem 4.12. First, recall that

$$\mathrm{Var}[Q|Y_1, ..., Y_n] = \frac{1}{\frac{1}{\sigma_Q^2} + \frac{n}{\sigma_\eta^2}} = \frac{\sigma_Q^2 \sigma_\eta^2}{\sigma_\eta^2 + n\sigma_Q^2}.$$

Solving for n_2 in the equation $\mathrm{Var}_1[Q|Y_1, ..., Y_{n_1}] = \mathrm{Var}_2[Q|Y_1, ..., Y_{n_2}]$,

$$\frac{\sigma_Q^2 \sigma_{\eta_1}^2}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2} = \frac{\sigma_Q^2 \sigma_{\eta_2}^2}{\sigma_{\eta_2}^2 + n_2\sigma_Q^2}$$

we get

$$\sigma_{\eta_1}^2(\sigma_{\eta_2}^2 + n_2\sigma_Q^2) = \sigma_{\eta_2}^2(\sigma_{\eta_1}^2 + n_1\sigma_Q^2)$$

and hence

$$\sigma_{\eta_1}^2 n_2 = \sigma_{\eta_2}^2 n_1.$$

Extracting n_2, we find that $n_2 = \frac{\sigma_{\eta_2}^2}{\sigma_{\eta_1}^2} n_1$. ■

Proof of Theorem 4.13. First, recall that

$$\mathbb{E}[Q|Y_1, ..., Y_n] = \mu_Q + \left[\frac{1}{\frac{\sigma_\eta^2}{\sigma_Q^2} + n}, \dots\right] \cdot (\mathbf{y} - \mu_y) = \mu_Q + \left[\frac{\sigma_Q^2}{\sigma_\eta^2 + n\sigma_Q^2}, \dots\right] \cdot (\mathbf{y} - \mu_y)$$

Now,

$$\mathbb{E}_1[Q|Y_1,...,Y_{n_1}] - \mathbb{E}_2[Q|Y_1,...,Y_{n_2}] =$$

$$\left[\frac{\sigma_Q^2}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2}, \dots\right] \cdot (\mathbf{y_1} - \mu_y) - \left[\frac{\sigma_Q^2}{\sigma_{\eta_2}^2 + n_2\sigma_Q^2}, \dots\right] \cdot (\mathbf{y_2} - \mu_y)$$

$$= \frac{\sigma_Q^2}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2} n_1(\bar{\mathbf{y_1}}) - \frac{\sigma_Q^2}{\sigma_{\eta_2}^2 + n_2\sigma_Q^2} n_2(\bar{\mathbf{y_2}})$$

$$= \frac{\sigma_Q^2 n_1}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2}(\bar{\mathbf{y_1}}) - \frac{\sigma_Q^2 n_2}{\sigma_{\eta_2}^2 + n_2\sigma_Q^2}(\bar{\mathbf{y_2}})$$

$$= \frac{\sigma_Q^2 n_1}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2}(\bar{\mathbf{y_1}}) - \frac{\sigma_Q^2 \frac{\sigma_{\eta_2}^2}{\sigma_{\eta_1}^2} n_1}{\sigma_{\eta_2}^2 + \frac{\sigma_{\eta_2}^2}{\sigma_{\eta_1}^2} n_1\sigma_Q^2}(\bar{\mathbf{y_2}})$$

$$= \frac{\sigma_Q^2 n_1}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2}(\bar{\mathbf{y_1}}) - \frac{\sigma_Q^2 n_1}{\sigma_{\eta_1}^2 + n_1\sigma_Q^2}(\bar{\mathbf{y_2}})$$

■

Chapter 5

Conclusion and Future Work

5.1 Generalization

In this thesis, we derived uniform convergence for multicalibration notion. We improved lower bounds from [117] on the sample size needed to guarantee uniform convergence of multicalibration for both finite and infinite predictor classes. For finite classes, the bounds now logarithmically depends on the size of the class (as in the case of uniform convergence for learning). As a result, a dependence of $\log(|\mathcal{H}|)$ is essential for the sample complexity, similar to lower bounds on sample complexity for agnostic PAC learning. For infinite classes, We have improved the lower bounds and show that it depends linearly in the Natarajan dimension of the class (as in the case of uniform convergence for multi-class learning).

Moreover, we have demonstrated how to apply the techniques to obtain uniform convergence of another commonly notions used in Data Science, F-scores, and in particular for an adjusted notion of F-Scores for subpopulations.

An interesting problem for future work is to enable an infinite number of subpopula-

tions defined by a class of binary functions with bounded VC-dimension. Deriving uniform convergence bounds in this setting will require overcoming some new challenges, since one cannot simply enumerate all subpopulations.

5.2 Reinforcement Learning

5.2.1 Dueling Teams

In the following we discuss several implications of our results as well as directions for future work.

Checking Condorcet winners beyond additive linear orders As we have briefly discussed within Section 3.1.5, the question how many duels are necessary to prove (or disprove) that a given team is a Condorcet winning team (even in an instance with $3k$ players) remains open for total orders that are not additive linear. A polynomial upper bound for this number would, together with our algorithm of Theorem 3.8, yield an algorithm with a polynomial number of duels. We formalize this observation within the following Corollary.

Corollary 5.1. *Let q be the number of duels required to check whether a given team is a Condorcet winning team within an instance with $\mathcal{O}(k)$ players. Then, there exists an algorithm that identifies a Condorcet winning team within $\mathcal{O}(kn\log(k)+k^2log(k)q)$ duels.*

Lower Bounds For the stochastic and the deterministic setting, there exists a lower bound of $\Omega(n-2k)$ duels in order to identify a Condorcet winning team: Consider an adversary that fixes, over time, a reverse lexicographical order, i.e., a duel is decided against the worst player participating. When the algorithm performs its first duel, the adversary picks an arbitrary player from the duel, makes him player n and answer

the query accordingly. Then, whenever the algorithm performs a duel containing a player which has already been fixed, the adversary decides the duel against the worst fixed player participating. Otherwise, he picks an arbitrary player from the duel and fixes him to become player $n-t$, where t is the number of so far fixed players. As long as $t < n - 2k$, the algorithm cannot not identify a Condorcet winning team.

Theorem 5.2. *Any algorithm that identifies a Condorcet winning team performs at least $n - 2k$ duels.*

Note that the above theorem is tight in the dependency on n, for small team size $k = o(n)$. Deriving tighter lower bounds for our team setting, especially the dependency on the team size, is an interesting question for future work.

Regret Bound In this section we provided algorithms to identify, with high probability, a Condorcet winning team. However, there exist other performance metrics for online learning theory, which apply in particular in MAB and dueling bandits.
As there exists more than a single Condorcet winning team, it is reasonable to define regret w.r.t. the best possible team, i.e., A_k^* for our setting, i.e.,

$$R_T = \sum_{t=1}^{T} \min \{P_{A_k^*,A_t} - 1/2, P_{A_k^*,B_t} - 1/2\},$$

where (A_t, B_t) is the selected duel at time t and T is the time horizon[1].
Using the second part of Theorem 3.7, one can choose $\delta = 1/(Tn)$ and achieve a regret bound of

$$R_T = (1-(Tn)^{-1})\cdot n(\Delta^{-2}(\log(T)+\log\log\Delta^{-1})+(Tn)^{-1} = \mathcal{O}(n(\Delta^{-2}(\log(T)+\log\log\Delta^{-1})).$$

[1]This definition is based on weak regret for dueling bandits, as defined in [134].

This follows from the SST of the distinguibilities (Lemma 3.4) implies $\Delta_i \geq \Delta$ for all $i \in [n]$.

5.2.2 Departing Bandits

This section introduces a MAB model in which the recommender system influences both the rewards accrued *and* the length of interaction. We dealt with two classes of problems: A single user type with general departure probabilities (Section 3.2.4) and the two user types, two categories where each user departs after her first no-click (Section 3.2.5). For each problem class, we started with analyzing the planning task, then characterized a small set of candidates for the optimal policy, and then applied Algorithm 11 to achieve sublinear regret.

In the full version [17], we also consider a third class of problems: Two categories, multiple user types ($M \geq 2$) where user departs with their first no-click. We use the closed-form expected return derived in Theorem 3.42 to show how to use dynamic programming to find approximately optimal planning policies. We formulate the problem of finding an optimal policy for a finite horizon H in a recursive manner. Particularly, we show how to find a $1/2^{O(H)}$ additive approximation in run-time of $O(H^2)$. Unfortunately, this approach cannot assist us in the learning task. Dynamic programming relies on skipping sub-optimal solutions to sub-problems (shorter horizons in our case), but this happens on the fly; thus, we cannot a-priori define a small set of candidates like what Algorithm 11 requires. More broadly, we could use this dynamic programming approach for more than two categories, namely for $K \geq 2$, but then the run-time becomes $O(H^K)$.

There are several interesting future directions. First, achieving low regret for the setup in Section 3.2.5 with $K \geq 2$. We suspect that this class of problems could enjoy a solution similar to ours, where candidates for optimal policies are mixing

two categories solely. Second, achieving low regret for the setup in Section 3.2.5 with uncertain departure (i.e., $\mathbf{\Lambda} \neq 1$). Our approach fails in such a case since we cannot use belief-category walks; these are no longer deterministic. Consequently, the closed-form formula is much more complex and optimal planning becomes more intricate. These two challenges are left open for future work.

5.2.3 SafeZone

In this section, we have introduced the SafeZone problem. We have shown the it is NP-hard even when the model is known, and designed a nearly $(2\rho, 2k^*)$ approximation algorithm for the case where the model and policy are unknown to the algorithm. Beyond improving the approximation factors (or showing that it cannot be done unless $P = NP$), a natural direction for future work is the following. Given $\rho > 0$ and an MDP (known or unknown to the learner), find a policy with a small $\rho-$safe set, with nearly optimal value. In fact, an efficient solution for this could pave the way to improve compactness of the policy representation. An interesting observation that comes up from the empirical demonstration is that different policies result in different sizes of SafeZones, and that the optimal policy does not necessarily has the smallest SafeZone.

5.3 Societal Challenges

5.3.1 Candidate Screening

Consider two groups with identically-distributed skills and characterized by different noise levels in screening. Our results demonstrate that if a regulatory body (e.g., policymakers or a regulator) insists on the same number of tests and the same decision rule for both groups, this would yield higher false positive rates in any threshold

policy. As a result, hired candidates from the noisier group would suffer higher rates of firing. In turn, this might lead employers to erroneously conclude that this group's skill level is lower than it actually is. This section presents a policy that handles this problem by minimizing the false positive rates of both groups, in the form of a greedy policy. Moreover, the greedy policy is efficient, minimizing the expected number of tests per hire among all policies that achieve a specified false positive rate and continue testing every candidates that appear better than the a new one. However, the dynamic policy will still suffer (as does the simple threshold policy) from higher false negative rates for the noisier group, violating a notion of fairness dubbed *equality of opportunity* in the recent literature on fairness in machine learning [63]. We addressed this problem by modifying the greedy policy to reject candidate iff $\Pr[y_i = +1|\hat{y}_{i,1} \dots \hat{y}_{i,\tau}] < \epsilon'$ by setting $\epsilon' < p$. Our greedy policy can be made forgiving and equalize false negative rates across groups.

Implications for Fairness When it comes to "business justification", Civil Rights regulation in the United States might be open to more than one interpretation regarding group-based disparities. In disparate impact doctrine, the statistical disparity of interest, e.g., in the famous 4/5 test concerns the decisions itself. In our model, if one were to apply a uniform hiring policy, administering the same number of tests to all applicants and applying the same threshold, a disparate impact might emerge. By subjecting members of noisier groups to more tests, we can equalize the confusion matrix entries across groups, seemingly eliminating any disparate impact concerning outcomes.

However, in this case, both the number of tests administered, and the inferences drawn from the results depend explicitly on group membership, potentially raising concerns about disparate treatment and procedural fairness. Another interesting

question might be to consider what disparate doctrine might have to say about disparities not in outcomes but in testing procedures.

Our setup motivates a new dimension to the discussion—even when members of the two groups have statistically identical outcomes, and even putting aside concerns about group-blindness, members of the more heavily-tested group may experience adversity. For example, perhaps these candidates, subject to more interviews, would not be able to interview with as many employers, thus lowering their overall likelihood of finding employment.

It would be interesting to introduce strategic behavior to our setting and understand the implications. For example, the candidates might have a utility that depends on whether they received the job, and disutility associated with how long their interview process was. Their overall utility can simply the difference between the two. Such a strategic model will cause some candidates not to apply, and the stream of candidates applying would have significant different characteristics than the overall population. Such a strategic setting would pose additional fairness challenges, since the mechanism would also control applies and not only who is hired.

Bibliography

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

[2] Dennis J Aigner and Glen G Cain. Statistical theories of discrimination in labor markets. *ILR Review*, 30(2):175–187, 1977.

[3] Alnour Alharin, Thanh-Nam Doan, and Mina Sartipi. Reinforcement learning interpretation methods: A survey. *IEEE Access*, 8:171058–171077, 2020.

[4] Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1168–1176, 2018.

[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[6] Kenneth Arrow et al. The theory of discrimination. *Discrimination in labor markets*, 3(10):3–33, 1973.

[7] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

[8] Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 2220–2228, 2013.

[9] Gal Bahar, Omer Ben-Porat, Kevin Leyton-Brown, and Moshe Tennenholtz. Fiduciary bandits. In *International Conference on Machine Learning*, pages 518–527. PMLR, 2020.

[10] Gal Bahar, Rann Smorodinsky, and Moshe Tennenholtz. Economic recommendation systems: One page abstract. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 757–757, New York, NY, USA, 2016. ACM.

[11] Maria-Florina F Balcan, Travis Dick, Ritesh Noothigattu, and Ariel D Procaccia. Envy-free classification. In *Advances in Neural Information Processing Systems 32*. 2019.

[12] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[13] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[14] Yahav Bechavod, Christopher Jung, and Zhiwei Steven Wu. Metric-free individual fairness in online learning, 2020.

[15] Gary S Becker. The economics of discrimination chicago. *University of Chicago*, 1957.

[16] Shai Ben-David, Nicolò Cesa-Bianchi, David Haussler, and Philip M. Long.

Characterizations of learnability for classes of {0, ..., n}-valued functions. *J. Comput. Syst. Sci.*, 1995.

[17] Omer Ben-Porat, Lee Cohen, Liu Leqi, Zachary C. Lipton, and Yishay Mansour. Modeling attrition in recommender systems with departing bandits. *arXiv preprint arXiv:2203.13423*, 2022.

[18] Omer Ben-Porat, Lee Cohen, Liu Leqi, Zachary C. Lipton, and Yishay Mansour. Modeling attrition in recommender systems with departing bandits. *Accepted to AAAI Conference on Artificial Intelligence (AAAI) 2022.*, 2022.

[19] Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, pages 1–108, 01 2021.

[20] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 2018.

[21] Guy Blanc, Jane Lange, and Li-Yang Tan. Provably efficient, succinct, and precise explanations. *Advances in Neural Information Processing Systems*, 34, 2021.

[22] Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 2007.

[23] Ulrik Brandes, Eugenia Holm, and Andreas Karrenbauer. Cliques in regular graphs and the core-periphery problem in social networks. In *International Conference on Combinatorial Optimization and Applications*, pages 175–186. Springer, 2016.

[24] Brian Brost, Yevgeny Seldin, Ingemar J Cox, and Christina Lioma. Multi-dueling bandits and their application to online ranker evaluation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 2161–2166, 2016.

[25] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[26] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

[27] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 258–265. PMLR, 2013.

[28] Róbert Busa-Fekete, Eyke Hüllermeier, and Adil El Mesaoudi-Paul. Preference-based online learning with dueling bandits: A survey. Technical report, arXiv:1807.11398, 2018.

[29] Junyu Cao, Wei Sun, Zuo-Jun Max Shen, and Markus Ettl. Fatigue-aware bandits for dependent click models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3341–3348, 2020.

[30] Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[31] Nicol‘o Cesa-Bianchi and G´abor Lugosi. *Prediction, learning, and games*. Cambridge Univ Press, 2006.

[32] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 2009.

[33] Lijie Chen, Jian Li, and Mingda Qiao. Towards instance optimal bounds for best arm identification. In *Proceedings of the 30th Conference on Learning Theory (COLT)*, pages 535–592. PMLR, 2017.

[34] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2014.

[35] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 2017.

[36] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[37] T. Anne Cleary. Test bias: Prediction of grades of negro and white students in integrated colleges. *Journal of Educational Measurement*, 1968.

[38] Lee Cohen, Zachary C. Lipton, and Yishay Mansour. Efficient candidate screening under multiple tests and implications for fairness. *CoRR*, abs/1905.11361, 2019.

[39] Lee Cohen, Zachary C. Lipton, and Yishay Mansour. Efficient candidate screening under multiple tests and implications for fairness. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, 2020.

[40] Lee Cohen and Yishay Mansour. Optimal algorithm for bayesian incentive-compatible. In *ACM Conf. on Economics and Computation (EC)*, 2019.

[41] Lee Cohen, Yishay Mansour, and Michal Moshkovitz. Finding safe zones of policies markov decision processes. *CoRR*, 2022.

[42] Lee Cohen, Ulrike Schmidt-Kraepelin, and Yishay Mansour. Dueling bandits with team comparisons. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[43] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[44] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[45] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In *Proceedings of the 24th Annual Conference on Learning Theory*, Proceedings of Machine Learning Research, 2011.

[46] Frank den Hollander, MV Menshikov, and SE Volkov. *Two problems about random walk in a random field of traps*. Citeseer, 1995.

[47] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012.

[48] Yonina Eldar and Gitta Kutyniok. *Compressed Sensing: Theory and Applications*. 01 2012.

[49] Yousef Emam, Paul Glotfelter, Zsolt Kira, and Magnus Egerstedt. Safe model-based reinforcement learning using robust control barrier functions. *CoRR*, 2021.

[50] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. In *Conference on Fairness, Accountability and Transparency (FAT*)*, 2018.

[51] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In Jyrki Kivinen and Robert H. Sloan, editors, *Computational Learning Theory, 15th Annual Conference on Computational Learning Theory, COLT 2002, Sydney, Australia, July 8-10, 2002, Proceedings*, 2002.

[52] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(39):1079–1105, 2006.

[53] Eyal Even-Dar and Yishay Mansour. Approximate equivalence of markov decision processes. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, 2003.

[54] Julius Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik*, 1902(124):1–27, 1902.

[55] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.

[56] Dean P. Foster and Sergiu Hart. Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 2018.

[57] Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, 1998.

[58] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16:1437–1480, 2015.

[59] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems 31*. 2018.

[60] Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 2003.

[61] H. D. Grossman. The twelve-coin problem. *Scripta Mathematica*, 11:360–361, 1945.

[62] Richard K. Guy and Richard J. Nowakowski. Coin-weighing problems. *The American Mathematical Monthly*, 102(2):164–167, 1995.

[63] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems (NeurIPS)*, 2016.

[64] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2015.

[65] Aria HasanzadeZonuzy, Archana Bura, Dileep M. Kalathil, and Srinivas Shakkottai. Learning with safety constraints: Sample complexity of reinforcement learning for constrained mdps. In *AAAI*, 2021.

[66] Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2018.

[67] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *CoRR*, 2021.

[68] Christopher Hillar and Andre Wibisono. Maximum entropy distributions on graphs, 2018.

[69] Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *World Wide Web Conference (WWW)*, 2018.

[70] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 2017.

[71] Christina Ilvento. Metric Learning for Individual Fairness. In *1st Symposium on Foundations of Responsible Computing (FORC 2020)*, 2020.

[72] Huiwen Jia, Cong Shi, and Siqian Shen. Multi-armed bandit with sub-exponential rewards. *Operations Research Letters*, 2021.

[73] Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. *arXiv preprint arXiv:1605.07139*, 2016.

[74] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[75] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 511–518. PMLR, 2010.

[76] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *International Conference on Data Mining (ICDM)*, 2010.

[77] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *ICDM Workshops*, 2011.

[78] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

[79] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2018.

[80] Michael Kearns and Aaron Roth. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, Inc., USA, 2019.

[81] Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, 2019.

[82] Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Fairness through computationally-bounded awareness. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 2018.

[83] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Leibniz International Proceedings in Informatics (LIPIcs), 2017.

[84] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2017.

[85] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[86] Nathan Korda, Balázs Szörényi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016.

[87] Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the wisdom of the crowd. *Journal of Political Economy*, 122:988–1012, 2014.

[88] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[89] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[90] Liu Leqi, Fatma Kilinc-Karzan, Zachary C Lipton, and Alan L Montgomery. Rebounding bandits for modeling satiation effects. *arXiv preprint arXiv:2011.06741*, 2020.

[91] Jeffrey Li, Vaishnavh Nagarajan, Gregory Plumb, and Ameet Talwalkar.

A learning theoretic perspective on local explainability. *arXiv preprint arXiv:2011.01205*, 2020.

[92] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2018.

[93] Lydia T. Liu, Max Simchowitz, and Moritz Hardt. The implicit fairness criterion of unconstrained learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[94] Yang Liu and Chien-Ju Ho. Incentivizing high quality user contributions: New arm generation in bandit learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[95] Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C. Parkes. Calibrated fairness in bandits. *CoRR*, 2017.

[96] Zhongqi Lu and Qiang Yang. Partially observable markov decision process for recommender systems. *CoRR*, abs/1608.07793, 2016.

[97] Kanak Mahadik, Qingyun Wu, Shuai Li, and Amit Sabne. *Fast Distributed Bandits for Online Recommendation Systems*. 2020.

[98] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *ACM Conf. on Economics and Computation (EC)*, 2015.

[99] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 2021.

[100] Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2488–2497, 06–11 Aug 2017.

[101] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[102] Michal Moshkovitz, Yao-Yuan Yang, and Kamalika Chaudhuri. Connecting interpretability and robustness in decision trees through separation. *arXiv preprint arXiv:2102.07048*, 2021.

[103] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Y Narahari. Achieving fairness in the stochastic multi-armed bandit problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5379–5386, 2020.

[104] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Knowledge Discovery in Databases (KDD)*, 2008.

[105] Andrzej Pelc. Searching games with errors - fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002.

[106] Samuel Pfrommer, Tanmay Gautam, Alec Zhou, and Somayeh Sojoudi. Safe reinforcement learning with chance-constrained model predictive control. *CoRR*, 2021.

[107] Edmund S Phelps. The statistical theory of racism and sexism. *The american economic review*, pages 659–661, 1972.

[108] Ciara Pike-Burke and Steffen Grünewälder. Recovering bandits. *arXiv preprint arXiv:1910.14354*, 2019.

[109] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In I. Guyon, U. V. Luxburg, S. Bengio,

H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*. 2017.

[110] Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

[111] Idan Rejwan and Yishay Mansour. Top-k combinatorial bandits with full-bandit feedback. In *Proceedings of the 31st Internationcal Conference on Algorithmic Learning Theory (ALT)*, pages 752–776. PMLR, 2020.

[112] Wenbo Ren, Jia Liu, and Ness Shroff. The sample complexity of best-k items selection from pairwise comparisons. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8051–8072. PMLR, 13–18 Jul 2020.

[113] Wenbo Ren, Jia Liu, and Ness B Shroff. PAC ranking from pairwise and listwise queries: Lower bounds and upper bounds. Technical report, arxiv.org/abs/1806.02970, 2018.

[114] Tom Ron, Omer Ben-Porat, and Uri Shalit. Corporate social responsibility via multi-armed bandits. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 26–40, 2021.

[115] Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 805–814. AUAI Press, 2018.

[116] Sven Schmit, Virag Shah, and Ramesh Johari. Optimal testing in the experiment-rich regime. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[117] Eliran Shabat, Lee Cohen, and Yishay Mansour. Sample complexity of uniform convergence for multicalibration. In *Advances in Neural Information Processing*

Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[118] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

[119] Guy Shani, David Heckerman, and Ronen I. Brafman. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(43):1265–1295, 2005.

[120] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019.

[121] Aleksandrs Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

[122] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Proceedings of 33rd the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[123] Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2018.

[124] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[125] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

[126] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, 2018.

[127] Kari Watkins, Calvin Clark, Patricia Mokhtarian, Giovanni Circella, Susan Handy, and Alison Kendall. *Bicyclist Facility Preferences and Effects on Increasing Bicycle Trips*. 2020.

[128] Ward Whitt. Uniform conditional stochastic order. *Journal of Applied Probability*, 17(1):112–123, 1980.

[129] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[130] Blake Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, Proceedings of Machine Learning Research, 2017.

[131] Wanqiao Xu, Kan Xu, Hamsa Bastani, and Osbert Bastani. Safely bridging offline and online reinforcement learning. *CoRR*, 2021.

[132] Gal Yona and Guy N. Rothblum. Probably approximately metric-fair learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

[133] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. volume 78, 01 2009.

[134] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538 – 1556, 2012.

[135] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. Jointly learning to recommend and advertise. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[136] Yifei Zhao, Yu-Hang Zhou, Mingdong Ou, Huan Xu, and Nan Li. Maximizing cumulative user engagement in sequential recommendation: An online optimization perspective. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[137] Yuan Zhou, Xi Chen, and Jian Li. Optimal pac multiple arm identification with applications to crowdsourcing. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 217–225. PMLR, 2014.

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע"ש בלבטניק

נושאים בלמידת מכונה, למידה מחיזוקים, וחברה

חיבור זה הוגש כחלק מהדרישה לקבלת התואר "דוקטור לפילוסופיה"

על ידי

לי כהן

בהנחייתה של פרופ' ישי מנצור

הוגש לסנאט של אוניברסיטת תל-אביב

אלול תשפ"ב

תמצית

בעשרות השנים האחרונות, התחום של למידת מכונה נהנה מהצלחה כבירה, ומהשפעה אדירה על חיי היום יום שלנו עם השלכות על מגוון רחב של אפליקציות כגון מערכות המלצה, רכבים אוטונומיים, ותרגום. למידה חישובית תיאורטית היא אבן נגף עיקרית למתודולוגיות נוכחיות ועתידיות של למידת מכונה. בתיזה הזו, אנחנו מרחיבים ומפתחים מתודולוגיות יעילות עבור תחומים של למידה מקוונת, הכללה, ומתן מענה לצרכים חברתיים כגון הוגנת

תוצאות התיזה פורסמו בחמישה מאמרים, שארבעה מתוכם הופיעו בכנסים הבאים: FORC'20, NeurIPS'20, NeurIPS'21, and AAAI'22.