3rd Symposium on Foundations of Responsible Computing

FORC 2022, June 6–8, 2022, Cambridge, MA, USA

Edited by

L. Elisa Celis

LIPIcs – Vol. 218 – FORC 2022 www.dagstuhl.de/lipics

Editors

L. Elisa Celis
Department of Statistics and Data Science, Yale University, New Haven, CT, USA
elisa.celis@yale.edu

ACM Classification 2012
Theory of computation; Security and privacy → Formal methods and theory of security; Security and privacy; Social and professional topics → Computing / technology policy; Theory of computation → Theory and algorithms for application domains; Theory of computation → Machine learning theory; Theory of computation → Algorithmic game theory and mechanism design; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Probability and statistics; Applied computing → Law, social and behavioral sciences; Computing methodologies → Machine learning

ISBN 978-3-95977-226-6

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-226-6.

Publication date
July, 2022

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://portal.dnb.de.

Digital Object Identifier: 10.4230/LIPIcs.FORC.2022.0

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Papers

3rd Symposium on Foundations of Responsible Computing (FORC 2022).
Editor: L. Elisa Celis
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

The Symposium on Foundations of Responsible Computing (FORC), now in its third year, is a forum for mathematically rigorous research in computation and society writ large. The Symposium aims to catalyze the formation of a community supportive of the application of theoretical computer science, statistics, economics, and other relevant analytical fields to problems of pressing and anticipated societal concern.

Topics include, but are not restricted to theoretical approaches to fairness in machine learning, including the investigation of definitions, algorithms, lower bounds, and tradeoffs; formal approaches to privacy, including differential privacy; computational and mathematical social choice, including apportionment and redistricting; economic incentives, including mechanism design for social good; metrics and implications of robustness, including formal methods for explainability; bias in the formation of, and diffusion in, social networks; and mathematical approaches bridging computer science, law, and ethics; mathematically rigorous work on societal problems that have not traditionally received attention in the theoretical computer science literature.

Twenty-four papers were selected to appear at FORC 2022, held on June 6-8, 2022 at the Harvard University Center for Mathematical Sciences and Applications in Cambridge, MA, USA. The twenty-four papers were selected by the program committee, with the help of additional expert reviewers, out of fifty-three submissions. FORC 2022 offered two submission tracks: archival-option (giving authors of selected papers the option to appear in this proceedings volume) and non-archival (in order to accommodate a variety of publication cultures, and to offer a venue to showcase FORC-relevant work that will appear or has recently appeared in another venue). Seven archival-option and seventeen non-archival submissions were selected for the program.

Thank you to the entire program committee and to the external reviewers for their hard work during the review process amid the continued challenging conditions of the pandemic. It has been an honor and a pleasure to work together with you to shape the program of this young conference. Finally, we would like to thank our generous sponsors at the Harvard Center of Mathematical Sciences and Applications (CSMA) for partial conference support.

Elisa Celis
New Haven, CT
April 30, 2022

3rd Symposium on Foundations of Responsible Computing (FORC 2022).
Editor: L. Elisa Celis

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Organizers

Program Committee

JAlfredo Viola

Amit Deshpande

Ashia Wilson

Cristobal Guzman

Deeparnab Chakrabarty

Elisa Celis (PC Chair)

Gireeja Ranade

Guy Rothblum

Ingal Talgam-Cohen

Jamie Morgenstern

Kobbi Nissim

Kunal Talwar

Pan Xu

Rachel Cummings

Rasmus Pagh

Ravi Kumar

Ruth Urner

Sampath Kannan

Seth Neel

Shuchi Chawla

Toshihiro Kamishima

Steering Committee

Avrim Blum

Cynthia Dwork

Shafi Goldwasser

Sampath Kannan

Jon Kleinberg

Kobbi Nissim

Toni Pitassi

Omer Reingold

Guy Rothblum

Salvatore Ruggieri

Salil Vadhan

Adrian Weller

Student Volunteers

Vijay Keswani

Anay Mehrotra

3rd Symposium on Foundations of Responsible Computing (FORC 2022).
Editor: L. Elisa Celis
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Controlling Privacy Loss in Sampling Schemes: An Analysis of Stratified and Cluster Sampling

Mark Bun
Department of Computer Science, Boston University, MA, USA

Jörg Drechsler
Institute for Employment Research, Nürnberg, Germany
Joint Program in Survey Methodology, University of Maryland, College Park, MD, USA

Marco Gaboardi
Department of Computer Science, Boston University, MA, USA

Audra McMillan
Apple, Cupertino, CA, USA

Jayshree Sarathy[1]
Harvard John A. Paulson School of Engineering and Applied Sciences, Boston, MA, USA

Abstract

Sampling schemes are fundamental tools in statistics, survey design, and algorithm design. A fundamental result in differential privacy is that a differentially private mechanism run on a *simple random* sample of a population provides stronger privacy guarantees than the same algorithm run on the entire population. However, in practice, sampling designs are often more complex than the simple, data-independent sampling schemes that are addressed in prior work. In this work, we extend the study of privacy amplification results to more complex, data-dependent sampling schemes. We find that not only do these sampling schemes often fail to amplify privacy, they can actually result in privacy degradation. We analyze the privacy implications of the pervasive cluster sampling and stratified sampling paradigms, as well as provide some insight into the study of more general sampling designs.

2012 ACM Subject Classification Security and privacy → Privacy protections

Keywords and phrases privacy, differential privacy, survey design, survey sampling

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.1

Funding Bun, Drechsler, Gaboardi and Sarathy were supported by Cooperative Agreement CB20ADR0160001 with the Census Bureau. The views expressed in this paper are those of the authors and not those of the U.S. Census Bureau or any other sponsor.
Mark Bun: Supported also by NSF grants CCF-1947889 and CNS-2046425.
Audra McMillan: Part of this work was done while AM was supported by a Fellowship from the Cybersecurity & Privacy Institute at Northeastern University and NSF grant CCF-1750640.

1 Introduction

Sampling schemes are fundamental tools in statistics, survey design, and algorithm design. For example, they are used in social science research to conduct surveys on a random sample of a target population. They are also used in machine learning to improve the efficiency and accuracy of algorithms on large datasets. In many of these applications, however, the datasets are sensitive and privacy is a concern. Intuition suggests that (sub)sampling a dataset before analysing it provides additional privacy, since it gives individuals plausible deniability about

[1] Corresponding author

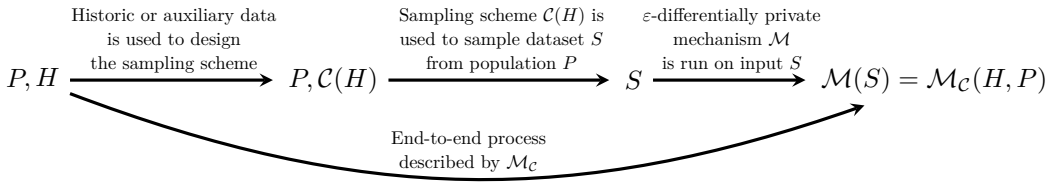

Figure 1 The structure of using a data-dependent sampling scheme.

whether their data was included or not. This intuition has been formalized for some types of sampling schemes (such as simple random sampling with and without replacement and Poisson sampling) in a series of papers in the differential privacy literature [23, 33, 11, 31]. Such *privacy amplification by subsampling* results can provide tight privacy accounting when analysing algorithms that incorporate subsampling, e.g. [32, 1, 21, 28, 19]. However, in practice, sampling designs are often more complex than the simple, data independent sampling schemes that are addressed in prior work. In this work, we extend the study of privacy amplification results to more complex and data dependent sampling schemes.

We consider the setting described in Figure 1. We have a *population* P and a historic or auxiliary data set H which is used to inform the sampling design. We think about the sampling scheme as a function $\mathcal{C}(H)$ of the historic or auxiliary data H. Using this sampling scheme, we draw a sample S from the population P, on which we run the differentially private mechanism $\mathcal{M}$. We can think about these multiple steps as comprising a mechanism $\mathcal{M}_{\mathcal{C}}(H, P)$ working directly on the population P and the historic data H whose privacy depends on the privacy of the mechanism $\mathcal{M}$ and on the properties of the sampling scheme $\mathcal{C}(H)$. While this is the general framework for the problem we study, we state the technical results in this paper for the simplified case where $H = P$; see Section 2.1 for further discussion.

1.1 Our contributions

We primarily focus on two classes of sampling schemes that are common in practice: *cluster sampling* and *stratified sampling*. In (single-stage) cluster sampling, the population arrives partitioned into disjoint clusters. A sample is obtained by selecting a small number of clusters at random, and then including all of the individuals from those chosen clusters. In stratified sampling, the population is partitioned into "strata." Individuals are then sampled at different rates according to which stratum they belong to.

For these more complex schemes, we find that privacy amplification can be negligible even when only a small fraction of the population is included in the final sample. Moreover, in settings where the sampling design is data dependent, privacy degradation can occur – some sampling designs can actually make privacy guarantees worse. Intuitively, this is because the sample design itself can reveal sensitive information. Our goal in this paper is to explain how and why these phenomena occur and introduce technical tools for understanding the privacy implications of concrete sampling designs.

Understanding randomised and data-dependent sampling. It is simple to show that deterministic, data-dependent sampling designs do not achieve privacy amplification, and can suffer privacy degradation. Motivated by this observation, we start by studying the privacy implications of randomised and data-dependent sampling, attempting to isolate their effects in the simplest possible setting.

Specifically, we aim to understand sampling schemes of the following form: For a possibly randomised function f (an "allocation rule"), sample $f(P)$ individuals uniformly from P without replacement. In Section 3, we study the case where f is randomised but data-independent, i.e., the number of individuals samples is drawn from a distribution that does not depend on P. We give an essentially complete characterization of what level of amplification is possible in terms of this distribution.

In Section 4, we turn our attention to data-dependent sampling. We identify necessary conditions for allocation rules f to enable privacy amplification by way of a hypothesis testing perspective; intuitively, for f to be a good amplifier, every differentially private algorithm must fail to distinguish the distributions of $f(P)$ and $f(P')$ for neighboring P, P'. We also study a specific natural allocation rule called *proportional allocation* that is commonly applied in stratified sampling. We design a simple randomised rounding method that offers a minor change to the way proportional allocation is generally implemented in practice, but that offers substantially better privacy amplification.

Cluster sampling. In Section 5, we study cluster sampling where a population partitioned into k clusters is sampled by selecting m clusters uniformly at random without replacement. Our results give tradeoffs between the privacy amplification achievable and the sizes of the clusters. In particular, privacy amplification is possible when all of the clusters are small. As the cluster sizes grow, the best achievable privacy loss rapidly approaches the baseline of the privacy guarantee of $\mathcal{M}$. We provide some insight into these results by connecting the privacy loss to the ability of a hypothesis test to determine from a differentially private output which clusters were included in the sample.

Stratified sampling. Building on our randomised rounding method for the "single-stratum" case, we show that stratified sampling with the proportional allocation rule amplifies privacy. Unfortunately, as in the single stratum case, there are natural lower bounds which limit extending this approach to other common allocation rules.

A common goal when choosing an allocation function $\boldsymbol{f}$ (a function which decides how many samples to draw from each stratum) is to minimise the variance of a particular statistic. For example, the popular Neyman allocation is the optimal allocation for computing the population mean. A natural question then is how to define and compute the optimal allocation when privacy is a concern? In this work, we will formulate the notion of an optimal allocation under privacy constraints. This formulation is somewhat subtle since the privacy implications of different allocation methods need to be properly accounted for. Our goal is to initiate the study of alternative allocation functions that may prove useful when privacy is a concern.

1.2 Related work

Several works have studied the privacy amplification of simple sampling schemes. Kasiviswanathan et al. [23] and Beimel et al. [9] showed that applying Poisson sampling before running a differentially private mechanism improves its end-to-end privacy guarantee. Subsequently, Bun et al. [11] analyzed simple random sampling with replacement in a similar way. Beimel et al. [10], Bassily et al. [7], and Wang et al. [34] analyzed simple random sampling without replacement. Imola and Chaudhuri [20] provide lower and upper bounds on privacy amplification when sampling from a multidimensional Bernoulli family, a task which has direct applications to Bayesian inference. Balle et al. [5] unified the analyses of privacy amplification of these mechanisms using the lenses of *probabilistic couplings*, an approach

that we also use in this paper. The effects that sampling can have on differentially private mechanisms is also studied from a different perspective in [13]. However, none of the prior works consider the privacy amplification of more complex, data-dependent sampling schemes commonly used in practice. To the best of our knowledge, this paper is the first to do so.

2 Background

2.1 Data-dependent sampling schemes

In the data-driven sciences, data is often obtained by sampling a fraction of the population of interest. This sample can be created in a wide variety of ways, referred to as the sample design. Sample designs can vary from simple designs such as taking a uniformly random subset of a fixed size, to more complex data-dependent sampling designs like cluster or stratified sampling. Data-dependent sampling designs achieve accuracy and meet budgeting goals by using historic or auxiliary data to exploit structure in the population. The privacy implications of simple random sampling are quite well understood from prior work. In this work, we will move beyond simple random sampling to analyse the privacy implications of more complex sampling designs, including data-dependent sampling.

An outline of the schema for data dependent sampling designs is given in Figure 1. There are ostensibly two datasets: H, the historic or auxiliary data that is used to design the sampling scheme $\mathcal{C}(H)$, and P, the current population that is sampled from. For the remainder of this paper, we make the simplifying assumption that $H = P$. That is, we will not distinguish between the historic or auxiliary data and the "current" data. Even if we only care about maintaining the privacy of the individuals in population P, this assumption is required if we have no information about the relationship between H and P. Thus, we view the function $\mathcal{M}_{\mathcal{C}}(P, H)$ as simply a function of P. We will refer to the size of the sample S as the *sample size*, and the fraction $|S|/|P|$ as the *sampling rate*.

More refined models can be obtained by imposing specific assumptions on the relationship between H and P, for example, by modeling the temporal correlation between historic and current data. We leave this for future work.

2.2 Differential privacy

Differential privacy (DP) is a measure of stability for randomised algorithms. It bounds the change in the distribution of the outputs of a randomised algorithm when provided with two datasets differing on the data of a single individual. We will call such datasets neighboring. In order to formalise what a "bounded change" means, we define (ε, δ)-indistinguishability. Two random variables P and Q over the same probability space are (ε, δ)-*indistinguishable* if for all sets of outcomes E over that probability space,

$$e^{-\varepsilon}(\Pr(Q \in E) - \delta) \leq \Pr(P \in E) \leq e^{\varepsilon} \Pr(Q \in E) + \delta.$$

If $\delta = 0$ then we will say that P and Q are ε-indistinguishable. For any $n \in \mathbb{N}$, let $\mathcal{U}^n$ be the set of all datasets of size n over elements of the data universe $\mathcal{U}$. Let $\mathcal{U}^* = \cup_{n \in \mathbb{N}} \mathcal{U}^n$ be the set of all possible datasets. We discuss two privacy definitions in this work corresponding to two different neighboring relations: *unbounded* differential privacy and *bounded* differential privacy. We will say two datasets are *unbounded neighbors* if one can be obtained from the other by adding or removing a single data point, and *bounded* neighbors if they have the same size, and one can be obtained from the other by changing the data of a single individual.

▶ **Definition 1.** *A mechanism $\mathcal{M} : \mathcal{U}^* \to \mathcal{O}$ is (ε, δ)-unbounded (resp. bounded) differentially private (DP) if for all pairs of unbounded (resp. bounded) neighboring datasets P and P', $\mathcal{M}(P)$ and $\mathcal{M}(P')$ are (ε, δ)-indistinguishable.*

We will use both bounded and unbounded DP throughout the paper as they are appropriate in different settings. When considering which notion to choose, it is important to consider which guarantees are meaningful in context. For example, it will be common in the sample designs we cover for the size of the sample S (see Figure 1) to be data-dependent. When considering these sampling designs, we will focus on mechanisms $\mathcal{M}$ that satisfy unbounded DP since bounded DP does not protect the sample size. However, bounded DP may be more appropriate for the privacy guarantee on $\mathcal{M}_{\mathcal{C}}$ in applications where it is unrealistic to assume that an individual can choose not to be part of the auxiliary dataset or the population. For example, the auxiliary data may be administrative data, data from a mandatory census, or data from a monopolistic service provider. Results and intuition are often similar between unbounded and bounded DP, although care should be taken when translating between the two notions. We note in particular that any ε-unbounded DP mechanism is 2ε-bounded DP.

2.3 Privacy amplification with uniform random sampling

Sampling does not provide strong differential privacy guarantees on its own. But when employed as a pre-processing step in a differentially private algorithm, it can amplify existing privacy guarantees. Intuitively, this is because if the choice of individuals is kept secret, sampling provides data subjects the plausible deniability to claim that their data was or was not in the final data set. This effect was first explicitly articulated in [29], and a formal treatment of the phenomenon was given in [5]. Three types of sampling are analysed in [5]: simple random sampling with replacement, simple random sampling without replacement, and Poisson sampling. In all three settings the privacy amplification is proportional to the probability of an individual not being included in the final computation. To gain some intuition before we move into the more complicated sampling schemes that are the focus on this paper, let us state and discuss the results from [5].

▶ **Theorem 2** ([5]). *Let $\mathcal{C}$ be a sampling scheme that samples m values out of n possible values without replacement. Given an (ε, δ)-bounded differentially private mechanism $\mathcal{M}$, we have that $\mathcal{M}_{\mathcal{C}}$ is (ε', δ')-bounded differentially private for $\varepsilon' = \log(1 + \frac{m}{n}(e^{\varepsilon} - 1))$ and $\delta' = \frac{m}{n}\delta$.*

To consider the implications of this result, notice that $\varepsilon' \leq \varepsilon$ for all values of $m \leq n$ so the sampled mechanism $\mathcal{M}_{\mathcal{C}}$ is strictly more private than the original mechanism $\mathcal{M}$. Further, taking into account the following two approximations which hold for small x,

$$e^x - 1 \approx x \tag{1}$$

$$\log(1 + x) \approx x, \tag{2}$$

we have that for small ε, $\varepsilon' \approx \frac{m}{n}\varepsilon$. So the degree of amplification in both parameters is roughly proportional to the sampling rate m/n.

2.4 How do people use subsampling amplification results?

Suppose we have a dataset that contains n records, and we want to estimate the proportion of individuals that satisfy some attribute in an ε-DP manner. Let us set our target privacy guarantee to be $\varepsilon = 1$. To do this, we can simply compute the proportion non-privately and

add Laplace noise with scale $1/n$. But, if we know that the dataset is a secret and simple random sample from a population of $100n$ individuals, then adding Laplace noise with scale $1/n$ as before will actually yield a stronger privacy guarantee of $\varepsilon' = 0.01$ for the underlying population. To get $\varepsilon' = 1$, we will need to add noise with scale only $1/(100n)$. In other words, the secrecy of the sample means that the computation has more privacy inherently, and therefore, we can add less noise in order to achieve the desired privacy guarantee.

Existing DP data analysis tools such as DP Creator [18, 17] employ privacy amplification results to provide better statistical utility. For example, the DP Creator interface prompts the user to input the population size if the data is a secret and random sample from a larger population of known size and take advantage of the resulting boost in accuracy without changing the privacy guarantee.

As we discussed before, privacy amplification results are also used to analyse algorithms that incorporate subsampling as one of their components. Privacy amplification results permit a tighter analysis of the privacy that these algorithm can guarantee. In particular, these algorithms are quite common in learning tasks, e.g. [32, 1, 21, 28, 19].

3 Randomised data-independent sampling rates

While we are ultimately interested in data-dependent sampling designs, we begin with an extension of Theorem 2 to non-constant but data-*independent* sampling rates. Prior results on privacy amplification by subsampling [23, 33, 11, 31, 6] all focus on constant sampling rates where the sampling rate (the fraction of the data set sampled) is fixed in advance. However, we will eventually see that randomising the sample rate is essential to privacy amplification when the target rate is data dependent. To work toward this eventual discussion, we first study the data-independent case to gain intuition for what properties of the distribution on sampling rates characterize how much privacy amplification is possible.

Suppose that there is a random variable t on $[n]$ and the sampling scheme is as follows: given a dataset P, a sample m is drawn from t, and then m subjects are drawn without replacement from P to form the sample S. In this section we consider unbounded differential privacy[2] for $\mathcal{M}$ and bounded differential privacy for $\mathcal{M}_{\mathcal{C}}$, where the total number of cases, n, is known and fixed. A simple generalisation of Theorem 2 immediately implies that the privacy loss of this randomised scheme is no worse than if t was concentrated on the maximum value in its support. However, prior work does not give insight into what happens when t is concentrated below its maximum or is evenly spread. What property of the distribution characterises its potential for privacy amplification? The following theorem characterizes the privacy amplification of sampling without replacement with data-independent randomised sampling rates.

▶ **Theorem 3.** *Let P be a dataset of size n, let t be a distribution over $\{0, 1, \ldots, n\}$, and let $\mathcal{C} : \mathcal{X} \to \mathcal{U}^*$ be the randomised, dataset-independent sampling scheme that randomly draws $m \sim t$ and samples m records from P without replacement. Define the distribution $\tilde{t}$ on $[n]$ where $\tilde{t}(m) \propto e^{\varepsilon m} \cdot t(m)$ for all $m \in [n]$.*

Upper bound: *Let $\mathcal{M} : \mathcal{U}^* \to \mathcal{O}$ be an ε-unbounded DP algorithm. Then, $\mathcal{M}_{\mathcal{C}}$ is ε'-bounded DP, where*

$$\varepsilon' = \log\left(1 + \frac{1}{n} \cdot \mathbb{E}_{m\sim\tilde{t}}[m] \cdot (e^{\varepsilon} - 1)\right).$$

[2] Note that we must use the unbounded differential privacy definition for $\mathcal{M}$ in this setting; otherwise, the sample size m would be fixed.

Lower bound: *There exists neighboring datasets P and P' of size n, and an ε-unbounded DP mechanism $\mathcal{M}$ such that if $\mathcal{M}_{\mathcal{C}}(P)$ and $\mathcal{M}_{\mathcal{C}}(P')$ are ε'-indistinguishable then*

$$\varepsilon' \geq -\log\left(1 - \frac{1}{n} \cdot \mathbb{E}_{m\sim\tilde{t}}[m] \cdot (1 - e^{-\varepsilon})\right)$$

First notice that Theorem 3 comports with the generalization of Theorem 2; as expected, if the support of t is contained within $[0, m']$ then $\mathbb{E}_{m\sim\tilde{t}}[m] \leq m'$, so the randomised scheme is at least as private as if t was concentrated on m'. It also determines that the property of t that determines the privacy amplification is $\mathbb{E}_{m\sim\tilde{t}}[m]$, the expectation of an exponential re-weighting of the distribution that gives more weight to larger sample sizes. When ε is small, the simple approximations $e^x - 1 \approx x$, $1 - e^{-x} \approx x$, and $\log(1 + x) \approx x$ mean that both the upper and lower bounds amount to

$$\varepsilon' \approx \frac{\mathbb{E}_{m\sim\tilde{t}}[m]}{n} \cdot \varepsilon.$$

Due to the exponential re-weighting,

$$\mathbb{E}_{m\sim\tilde{t}}[m] = \frac{\sum_{m=0}^{n} e^{\varepsilon m} \Pr(t = m) m}{\sum_{m=0}^{n} e^{\varepsilon m} \Pr(t = m)}$$

rapidly approaches n as the weight of t on values close to n increases. Intuitively, this means that even a small probability of sampling the entire dataset can be enough to ensure that there is no privacy amplification, even if the mode of t is much smaller than n. Conversely, if t is a light tailed distribution (say, subgaussian) concentrated on a value much smaller than n, then privacy amplification is possible.

For example, suppose that t is a truncated Gaussian on $[0, n]$ with mean $n/2$ and standard deviation σ. If t is highly concentrated then we expect the privacy guarantee of $\mathcal{M}_{\mathcal{C}}$ to be $\approx \varepsilon/2$. As σ grows we expect the privacy guarantee to tend towards ε as more weight is placed near n. In Figure 2, we illustrate the bounds of Theorem 3 numerically with this Gaussian example. We can see that when $n = 10,000$ and $\sigma \approx 800$, the privacy guarantee of $\mathcal{M}_{\mathcal{C}}$ is already close to $\varepsilon = 0.01$, the privacy guarantee of $\mathcal{M}$.

4 Data-dependent sampling rates

We now turn our attention to sampling schemes where sampling rates may depend on the data. The results in this section are motivated by *stratified sampling*, where the population is stratified into k disjoint sub-populations called strata, and an allocation function is used to determine how many samples to draw from each stratum. We will discuss stratified sampling with $k > 1$ in Section 6, but for simplicity and clarity, we first focus on the "single stratum" case. In this section, we develop tools and statements that we expect to be more broadly useful in understanding complex sampling designs.

Specifically, we consider the sampling design where one selects a number of cases according to a data-dependent function, and then samples that many cases via simple random sampling. That is, let $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ be a possibly randomised function and let $\mathcal{C}_f$ be the sampling function that on input P samples $f(P)$ data points uniformly without replacement from P. If $\mathcal{M}$ is an ε-DP algorithm, then how private is $\mathcal{M}_{\mathcal{C}_f}$?

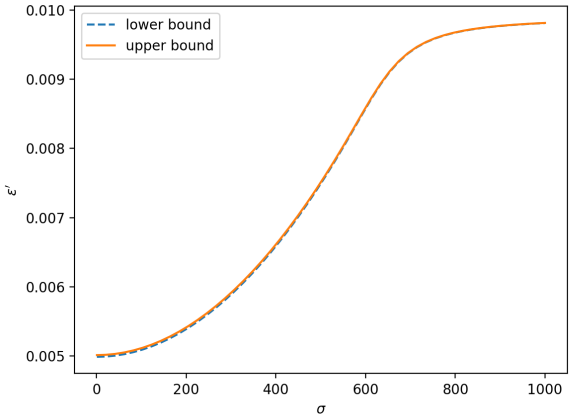

Figure 2 Numerical computation of the upper and lower bounds from Theorem 3 when t is truncated Gaussian supported on $[0, n]$ with mean $n/2$, where $n = 10^4$ and standard deviation σ varies from 1 to 10^3. The privacy parameter of the mechanism $\mathcal{M}$ is 0.01.

4.1 Sensitivity and privacy degradation

We first observe that if the function f used to determine sample size is highly sensitive, then privacy *degradation* may occur. That is, if the number of cases sampled may change dramatically on neighboring populations, then the output of a DP mechanism can immediately be used to distinguish between those populations. For example, suppose P and P' are neighboring populations, and f is a function where $f(P) = m$ and $f(P') = m + \Delta$. (That is, the local sensitivity of f at P is at least Δ.) Consider the ε-DP algorithm $\mathcal{M}^{\text{count}}$ that, on input a sample S, outputs the noisy count $|S| + \text{Lap}(1/\varepsilon)$ of the number of cases in the sample. Then $\mathcal{M}^{\text{count}}_{\mathcal{C}_f}(P)$ is distributed as $m + \text{Lap}(1/\varepsilon)$ whereas $\mathcal{M}^{\text{count}}{}_{\mathcal{C}_f}(P')$ is distributed as $m + \Delta + \text{Lap}(1/\varepsilon)$. When $\Delta \gg 1$, these distributions are far apart; the privacy loss between these two populations is $\Delta \cdot \varepsilon \gg \varepsilon$.

Thus, a *necessary* condition for achieving privacy amplification (rather than degradation) is that the function f has low sensitivity. In the following sections, we explore other conditions on low sensitivity functions that are necessary and sufficient for amplification.

4.2 Data dependent sampling and hypothesis testing

We established in the previous section that using a deterministic function to determine sample size results in privacy degradation. This raises the question: how much randomness is necessary to ensure privacy control? That is, what can we say about a randomised function $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ with the property that $\mathcal{M}_{\mathcal{C}_f}$ is ε'-DP for every ε-DP mechanism $\mathcal{M}$? In this section we establish a connection between the amplification properties of a function $\tilde{f}$ and hypothesis testing.

A simple hypothesis testing problem is specified by two distributions X and Y. A hypothesis test H for this problem attempts to determine whether the samples given as input are drawn i.i.d from X or from Y. If a hypothesis test is only given a single sample then we define the advantage of H to be

$$\text{adv}(H; X, Y) = \Pr_{m \sim X}[H(m) = X] - \Pr_{m \sim Y}[H(m) = X].$$

That is, the advantage is a measure of how likely the hypothesis test H is to correctly guess which distribution the sample was drawn from. The closer the advantage is to 1, the better the test is at distinguishing X from Y.

One common explanation of differential privacy is that an algorithm is differentially private if it is impossible to confidently guess from the output which of two neighbouring datasets was the input dataset. This interpretation can be formalised, following [35], by noting that if $\mathcal{M}$ is ε-DP and P and P' are neighbouring populations then for every hypothesis test H,

$$\mathrm{adv}(H; \mathcal{M}(P), \mathcal{M}(P')) \leq e^{\varepsilon} - 1 \approx \varepsilon.$$

We can establish a similar bound and interpretation of what it means for $\tilde{f}$ to amplify or preserve privacy. Suppose that $\tilde{f}$ is such that $\mathcal{M}_{\mathcal{C}_{\tilde{f}}}$ is ε'-DP for every ε-DP mechanism $\mathcal{M}$. Then in particular, for every ε-DP hypothesis test H, we have that $H(\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P))$ and $H(\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P'))$ are ε'-indistinguishable. Now, if we consider only hypothesis tests $H : \mathbb{N} \to \{\tilde{f}(P), \tilde{f}(P')\}$ that simply look at the size of the sample $\mathcal{C}_{\tilde{f}}(\cdot)$, then we can formalise this statement in the following way.

▶ **Proposition 4.** *Suppose $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ is such that for all ε-DP mechanisms $\mathcal{M}$, we have that $\mathcal{M}_{\mathcal{C}_{\tilde{f}}}$ is ε'-DP. Then for all neighboring datasets P, P', we have*

$$\max \mathrm{adv}(H; \tilde{f}(P), \tilde{f}(P')) \leq e^{\varepsilon'} - 1,$$

where the optimisation is over all hypothesis tests H such that for all $x \in \mathbb{N}$, and $b \in \{0, 1\}$, $e^{-\varepsilon} \Pr(H(x) = b) \leq \Pr(H(x + 1) = b) \leq e^{\varepsilon} \Pr(H(x) = b)$.

This result helps us build intuition for what type of survey designs could possibly amplify privacy. If $\tilde{f}$ results in privacy amplification then for any pair of neighbouring populations P and P', the distributions $\tilde{f}(P)$ and $\tilde{f}(P')$ must be close enough that they can not be distinguished between by any hypothesis test H such that $\log H$ is ε-Lipschitz. From this perspective the result in Section 4.1 follows from the fact that if $\tilde{f}$ is deterministic with high sensitivity then we can define an appropriate hypothesis test with large advantage based on $\mathcal{M}^{\mathrm{count}}$. This is a useful perspective to keep in mind throughout the remainder of the paper.

One consequence of this perspective is a lower bound on how well we can emulate a desired deterministic function f while controlling or amplifying privacy. Suppose that absent privacy concerns, an analyst has determined that they want to use a function f to determine the sample size. However, to avoid privacy degradation they replace f with a randomised function $\tilde{f}$. How close can $\tilde{f}$ get to f while maintaining or amplifying the original privacy level? We can obtain a lower bound on expected closeness of $f(P)$ and $\tilde{f}(P)$ by relating it to the well studied problem of estimation lower bounds in differential privacy.

▶ **Proposition 5.** *Let $f : \mathcal{U}^* \to \mathbb{R}$ and $\varepsilon, \varepsilon' > 0$. Suppose $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ is a randomised function such that for all ε-unbounded DP mechanisms $\mathcal{M}$, it holds that $\mathcal{M}_{\mathcal{C}_{\tilde{f}}}$ is ε'-bounded DP. If $\alpha \geq 0$ is such that for every ε'-unbounded DP mechanism $\mathcal{A}$, there exists a dataset P such that $\mathbb{E}[|\mathcal{A}(P) - f(P)|^2] \geq \alpha$, then there exists a dataset P such that*

$$\mathbb{E}[|\tilde{f}(P) - f(P)|^2] \geq \alpha - \left(\frac{1}{\varepsilon}\right)^2.$$

Proof. Define $\mathcal{M}_{SS} : \mathcal{U}^* \to \mathbb{N}$ as follows. For all $P \in \mathcal{U}^*$, $\mathcal{M}(P) = |P| + \mathrm{Lap}(1/\varepsilon)$. Then $\mathcal{M}$ is ε-unbounded DP. Suppose that $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ is such that for all ε-unbounded DP mechanisms $\mathcal{A}$, $\mathcal{A}_{\mathcal{C}_{\tilde{f}}}$ is ε'-bounded DP. This implies that $\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) = \tilde{f}(P) + \mathrm{Lap}(1/\varepsilon)$ is

ε'-bounded DP. Therefore, by the definition of α, there exists a population P such that $\sup_{P\in\mathcal{U}^n} \mathbb{E}[|\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) - f(P)|^2] \geq \alpha$. Also

$$\alpha \leq \mathbb{E}[|\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) - f(P)|^2] = \mathbb{E}[|\tilde{f}(P) + \mathrm{Lap}(1/\varepsilon) - f(P)|^2] = \mathbb{E}[]|\tilde{f}(P) - f(P)|^2] + (1/\varepsilon)^2.$$

After a small amount of rearranging we arrive at the result. ◀

The problem of lower bounding differentially private function estimation is well-studied [30, 4] in the privacy literature. The lower bounds essentially arise from the fact that $\mathcal{A}(P)$ and $\mathcal{A}(P')$ must be similar distributions for neighbouring databases, even if $f(P)$ and $f(P')$ are far apart. Since we know from Proposition 4 that $\tilde{f}(P)$ and $\tilde{f}(P')$ must also be close, we obtain the related lower bound. The slackness of $(1/\varepsilon)^2$ is a result of the fact that while $\mathcal{A}(P)$ and $\mathcal{A}(P')$ must be indistinguishable with respect to *any* hypothesis test, $\tilde{f}(P)$ and $\tilde{f}(P')$ need only be indistinguishable with respect to any ε-DP hypothesis test.

4.3 Privacy amplification from randomised rounding

Many functions used to determine data-dependent sampling rates have high sensitivity, but at least one common sampling method has low sensitivity: proportional sampling. In proportional sampling, a constant, data-independent fraction of the population is sampled independently from each stratum. This method is similar to simple random sampling, but a small amount of data dependence is introduced by the fact that the total number of samples across all strata must be an integer. In this section, we will show that while naïve implementations of proportional sampling can result in privacy degradation, a minor change in the sampling size function results in privacy amplification comparable to that afforded by simple random sampling.

Let $r \in [0,1]$ and $f(P) = r|P|$ for some constant $r \in (0,1)$. Since the output space of f is not $\mathbb{N}$, in practice, this is typically replaced with the deterministic function $\tilde{f}_{\mathrm{det},r}(P) = \mathrm{round}(r|P|)$, where $\mathrm{round}(\cdot)$ rounds its input to the nearest integer. Unfortunately, deterministic rounding can be problematic for privacy, as we can see through a simple example. Suppose P and P' are neighbouring populations such that $|P| = 14, |P'| = 15$, and $r = 1/10$. Then, deterministic rounding always results in one case being sampled from P and two cases being sampled from P'. As discussed in Section 4.1, such a data-dependent deterministic function can never result in privacy amplification.

We propose a simple and practical change to the rounding process that *does* guarantee roughly the expected level of privacy amplification. We replace the ideal function f with a randomised rounding function $\tilde{f}_{\mathrm{rand},r}$. That is, let $p = r|P| - \lfloor r|P| \rfloor$ so $\tilde{f}_{\mathrm{rand},r}(P) = \lceil r|P| \rceil$ with probability p, and $\tilde{f}_{\mathrm{rand},r}(P) = \lfloor r|P| \rfloor$ with probability $1-p$. The following proposition shows that, up to a constant factor, randomised rounding recovers the expected factor of r in privacy amplification.

▶ **Theorem 6** (Privacy Amplification from Randomised Rounding). *Let $r \in (0,1)$. Then for every ε-unbounded DP mechanism $\mathcal{M}$, the mechanism $\mathcal{M}_{\mathcal{C}_{\tilde{f}_{rand,r}}}$ is ε'-unbounded DP when restricted to datasets of size at least $1/r$, where $\varepsilon' = \log\big(1 + 2r(e^{2\varepsilon} - 1)\big) + \log(1 + r(e^{2\varepsilon} - 1)) \approx 6r\varepsilon$.*

The approximation at the end of the proposition follows from applying (1) and (2), which give that $\log(1 + 2r(\exp(2\varepsilon) - 1)) \approx 2r \cdot 2\varepsilon$ and $\log(1 + r(\exp(2\varepsilon) - 1)) \approx r \cdot 2\varepsilon$. The constant 6 can perhaps be optimized through a more careful analysis. Randomised rounding is a practical modification since it does not change the size of the sample very much; if traditional proportional allocation would typically assign m samples, then the modified algorithm allocates at most $m + 1$.

5 Cluster sampling

In cluster sampling, the population is partitioned into disjoint subsets, called clusters. A subset of the clusters is sampled and data subjects are selected from within the chosen clusters. If the sampling scheme uses a single stage design, all data subjects contained in the selected clusters will be included in the sample. Otherwise, a random sample of data subjects might be selected from each of the selected clusters (multi-stage design). Cluster sampling produces accurate results when the clusters are mutually homogeneous; that is, when the distributions within each cluster are similar to the distribution over the entire population.

In the survey context, cluster sampling is often performed due to time or budgetary constraints which make sampling many units from a few clusters cheaper and/or faster than sampling a few units from each cluster. A typical example in the survey context is when clusters are chosen to be geographic regions. Sampling a few geographic clusters and interviewing everybody in those clusters saves traveling costs compared to interviewing the same number of people based on a simple random sample from the population. In algorithm design, cluster sampling is often performed to improve the performance and accuracy of classifiers. In this setting, sampling often involves a two-step approach where the data is first clustered, using some clustering classifier, and then a subset of the clusters is selected. Forms of cluster samplings have been applied in several learning areas, for example in federated learning [16] and active learning [27].

5.1 Privacy implications of single-stage cluster sampling with simple random sampling

We focus here on a simple cluster sampling design that is commonly used in survey sampling and which naïvely appears to be a good candidate for privacy amplification: simple random sampling without replacement of clusters. That is, suppose the dataset P is divided into k clusters,

$$P = C_1 \sqcup \cdots \sqcup C_k$$

and the sampling mechanism $\mathcal{C}_\ell : \mathcal{U}^* \to \mathcal{U}^*$ chooses a random subset $I \subset [k]$ of size $\ell < k$, then maps P to $\sqcup_{i \in I} C_i$.

Since simple random sampling at the individual level provides good privacy amplification, one might expect the same to happen when the clusters are sampled in a similar way. In fact, this is true when the size of each cluster is small. However, if the clusters are large this sampling design achieves less amplification than might be expected. This is characterized by the following theorem showing a lower bound in this setting.

▶ **Theorem 7** (Lower Bound on Privacy Amplification for Cluster Sampling). *For any sequence $n_i > 0$ and privacy parameter $\varepsilon > 0$, there exist neighboring populations $P = C_1 \sqcup \cdots C_i \sqcup \cdots \sqcup C_k$ and $P' = C_1 \sqcup \cdots C_i' \sqcup \cdots \sqcup C_k$ (with $|C_i| = n_i$ and $C_i' = C_i \cup \{x\}$ for some $x \in \mathcal{U}$) and an ε-unbounded DP mechanism $\mathcal{M}$ such that if $\mathcal{M}_{\mathcal{C}_\ell}(P)$ and $\mathcal{M}_{\mathcal{C}_\ell}(P')$ are ε'-indistinguishable then*

$$\varepsilon' \geq \ln\left(1 + \frac{\frac{\ell}{k}}{\left(\frac{\ell}{k} + \left(1 - \frac{\ell}{k}\right) e^{-(n_i + n_{\min})\varepsilon}\right)} (e^\varepsilon - 1)\right),$$

where $n_i = |C_i|$ and $n_{\min} = \min_{j \in \{1,\cdots,i-1\} \cup \{i+1,\cdots,k\}} n_j$.

We can compare the expression in the theorem above with the one we have for simple random sampling without replacement (cf. Theorem 14 from [6]):

$$\varepsilon' = \ln\left(1 + \frac{m}{n}(e^{\varepsilon} - 1)\right),$$

where m samples are drawn from a population of size n. We see that the two expressions coincide if $n_i + n_{\min} = 0$, which is an unrealistic corner case. Let us instead consider the case in which all the clusters are small. In this case, the quantity $n_i + n_{\min}$ will also be small, and if $\varepsilon < 1$, we can still expect some privacy amplification. However, as the clusters grow in size, the quantity $n_i + n_{\min}$ will also increase, and the lower bound converges very quickly to ε, giving essentially no amplification.

Next, we present a corresponding upper bound.

▶ **Theorem 8** (Upper Bound on Privacy Amplification for Cluster Sampling)**.** *For any sequence $n_i > 0$, privacy parameter $\varepsilon > 0$, ε-unbounded DP mechanism $\mathcal{M} : \mathcal{U}^* \to \mathcal{O}$, and pair of neighboring populations P and P' such that $P = C_1 \sqcup \cdots C_i \sqcup \cdots \sqcup C_k$ and $P' = C_1 \sqcup \cdots C_i' \sqcup \cdots \sqcup C_k$ (with $|C_i| = n_i$ and $C_i' = C_i \cup \{x\}$ for some $x \in \mathcal{U}$), the mechanisms $\mathcal{M}_{\mathcal{C}_\ell}(P)$ and $\mathcal{M}_{\mathcal{C}_\ell}(P')$ are ε'-indistinguishable where*

$$\varepsilon' \leq \ln\left(1 + \frac{\frac{\ell}{k}}{\left(\frac{\ell}{k} + \left(1 - \frac{\ell}{k}\right) e^{-(n_i + n_{\max})\varepsilon}\right)}(e^{\varepsilon} - 1)\right),$$

and $n_{\max} = \max_{j \in \{1,\cdots,i-1\} \cup \{i+1,\cdots,k\}} n_j$,

Once again it is worth comparing the expression in the theorem above with the one we have for simple random sampling without replacement:

$$\varepsilon' = \ln\left(1 + \frac{m}{n}(e^{\varepsilon} - 1)\right).$$

Similar to the lower bound, the upper bound will quickly approach ε if the quantity $n_i + n_{\max}$ is large. If each cluster contains a single data point, the two bounds are close. This is not surprising since in this case the type of cluster sampling we considered is just simple random sampling without replacement. Note that while ℓ/k is the fraction of clusters included in the final sample and m/n is the fraction of data points, these are approximately the same when the clusters are small. If all the clusters are the same size, then $n_{\max} = n_{\min}$ and the upper and lower bounds we gave above match. The proofs of these results are contained in the Appendix.

5.2 Discussion and hypothesis testing

Privacy amplification by subsampling is often referred to as *secrecy of the sample* due to the intuition that the additional privacy arises from the fact that there is uncertainty regarding which user's data is in the sample. The key intuition then for Theorem 7 is that the larger the clusters are, the easier it is for a differentially private algorithm $\mathcal{M}$ to reverse engineer which clusters were sampled, breaking secrecy of the sample. Intuitively, if the clusters are different enough that a private algorithm can guess which clusters were chosen as part of the sample, then any amplification due to *secrecy of the sample* is negligible. We can formalize this intuition using once again using the lens of hypothesis testing. Note the framing in this section differs slightly from the framing in Section 4, although the underlying idea in both settings is that if a particular hypothesis test is effective, then there is a lower bound on the privacy parameter. In addition, note that privacy is also conserved in this setting, as $\mathcal{M}_{\mathcal{C}_\ell}$ is at least as private as $\mathcal{M}$. The question is: when is $\mathcal{M}_{\mathcal{C}_\ell}$ *more* private than $\mathcal{M}$?

▶ **Theorem 9.** *Let $\varepsilon > 0$, $\ell \in [0, k]$, $\mathcal{M} : \mathcal{U}^* \to \mathcal{O}$ be ε-DP and the sampling mechanism $\mathcal{C}_\ell$ be as defined in Section 5.1. Suppose there exists a hypothesis test $\mathcal{H} : \mathcal{O} \to \{0, 1\}$ such that*

$$\Pr(\mathcal{H}(\mathcal{M}_{\mathcal{C}_\ell}(P)) = 0 \mid C_i \in \mathcal{C}_\ell(P)) \geq e^{\varepsilon'} \Pr(\mathcal{H}(\mathcal{M}_{\mathcal{C}_\ell}(P)) = 0 \mid C_i \notin \mathcal{C}_\ell(P)).$$

Then there exists an event E in the output space of $\mathcal{M}$ such that for any neighboring population P' that differs from P in C_i, if

$$\varepsilon'' = \log \frac{\Pr(\mathcal{M}_{\mathcal{C}_\ell}(P) \in E | C_i \in \mathcal{C}_\ell(P))}{\Pr(\mathcal{M}(\mathcal{C}_\ell(P')) \in E | C_i \in \mathcal{C}_\ell(P'))} \in [0, \varepsilon],$$

and $\mathcal{M}_{\mathcal{C}_\ell}(P)$ and $\mathcal{M}_{\mathcal{C}_\ell}(P')$ are $\tilde{\varepsilon}$-indistinguishable, then

$$\tilde{\varepsilon} \geq \log\left(1 + (e^{\varepsilon''} - 1)\frac{\ell/k}{\ell/k + e^{-\varepsilon'}(1 - \ell/k)}\right).$$

The key take-away of this theorem is that for any ε-DP mechanism $\mathcal{M}$, if there exists a hypothesis test that, when given the output of $\mathcal{M}_{\mathcal{C}_\ell}(P)$, can confidently decide whether cluster C_i was chosen as part of the final sample, then the privacy guarantee of $\mathcal{M}_{\mathcal{C}_\ell}$ is no better than the privacy guarantee would be if we knew for certain that C_i was chosen as part of the sample. That is, in this setting, we gain no additional privacy as a result of secrecy of the sample. The parameter ε' controls how well the hypothesis test can determine whether $C_i \in \mathcal{C}_\ell$. As ε' increases, $\tilde{\varepsilon}$ approaches ε'', the privacy parameter if C_i is known to be part of the sample, so privacy amplification is negligible.

This view is consistent with Theorem 7. Consider a population where only data points in cluster i have a particular property and let $\mathcal{M}$ is an ε-DP mechanism that attempts to count how many data points with the property are in the final sample. If cluster i is large, then it is easy to determine from the output of the mechanism whether C_i is in the final sample. This example required cluster i to be distinguishable from the remaining clusters using a private algorithm. While examples as extreme as the one above may be uncommon in practice, clusters being different enough for a private algorithm to distinguish between them is not an unrealistic assumption.

In Section 5.1, we analysed a single stage design. All subjects contained in the selected clusters were included in the sample. In practice, multi-stage designs are common, where a random sample of subjects are selected from within each chosen cluster. If the sampling within each cluster is sufficiently simple then the privacy amplification from this stage can be immediately incorporated into the upper bound in Theorem 8. For example, if each subject within the chosen clusters is sampled with probability r and $\mathcal{M}$ is ε-DP, i.e., we perform Poisson sampling with probability r, then we immediately obtain an upper bound that is approximately $r\varepsilon$. One can also imagine more complicated schemes for selecting the chosen clusters. If these designs depend on properties of the data, then they are likely to result in privacy degradation. We leave this study for future work.

6 Stratified sampling

Finally, we turn our attention to another common sampling design: stratified sampling. In stratified sampling, the data is partitioned into disjoint subsets, called strata. A subset of data points is then sampled from each stratum to ensure the final sample contains data points from every stratum. Stratified sampling is common in survey sampling where it is used to improve accuracy and to ensure sufficient representation of sub-populations of interest. A classic use case of stratified sampling is business surveys, where businesses are typically

stratified by industry and number of employees, or by similar measures of establishment size. Stratification by establishment size results in substantial gains in accuracy compared to simple random sampling, while stratification by industry ensures that reliable estimates can be obtained at the industry level. Stratified sampling has several other applications; for example it is used in algorithm design to improve performance [2, 24], in private query design and optimization to improve accuracy [8], and to improve search and optimizations [25].

We focus here on *one-stage stratified sampling* using simple random sampling without replacement within each stratum to select samples. We also assume that the stratum boundaries have been fixed in advance. Given a target sample size m, the only design choice in this model is the *allocation function*, which determines how many samples to take from each stratum. Different allocation functions are used in practice. Which method is selected depends on the goals to be achieved (for example, ensuring constant sampling rates across strata or minimizing the variance for a statistic of interest).

Before we describe allocation functions in detail, let us establish some notation for stratified sampling. Suppose there are k strata in the population, and that each data point is a pair (s, x) where $s \in [k]$ denotes which stratum the data subject belongs to, and $x \in \mathcal{U}$ denotes their data. Let $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_k) : ([k] \times \mathcal{U})^* \to \mathbb{N}^k$ denote the allocation rule, so $\boldsymbol{f}_i(P)$ samples are drawn uniformly at random without replacement from the ith stratum, $P_i = \{(s, x) \in P \mid s = i\}$. The final sample S is the union of the samples from all the strata.

An important feature of stratified sampling is that the sampling rates can vary between the strata. This means that data subjects in strata with low sampling rates may expect a higher level of privacy than data subjects in strata with high sampling rates. This leads us to define a variant of differential privacy that allows the privacy guarantee to vary between the strata. This generalisation of differential privacy is tailored to stratified datasets and allows us to state more refined privacy guarantees than the standard definition is capable of.

▶ **Definition 10.** *Let $k \in \mathbb{N}$ and suppose there are k strata. A mechanism $\mathcal{A}$ satisfies $(\varepsilon_1, \cdots, \varepsilon_k)$-stratified bounded differential privacy if for all datasets P, data points (s, x) and (s', x'), $\mathcal{A}(P \cup \{(s, x)\})$ and $\mathcal{A}(P \cup \{(s', x')\})$ are $\max\{\varepsilon_s, \varepsilon_{s'}\}$-indistinguishable. The mechanism $\mathcal{A}$ satisfies $(\varepsilon_1, \cdots, \varepsilon_k)$-stratified unbounded differential privacy if for all datasets P, data points (s, x), $\mathcal{A}(P)$ and $\mathcal{A}(P \cup \{(s, x)\})$ are ε_s-indistinguishable.*

This definition is an adaptation of personalized differential privacy [22, 14, 3]. Note that it protects not only the *value* of an individual's data point, but also which stratum they belong to.

6.1 Optimal allocation with privacy constraints

In this section, we will discuss how to think about choosing an allocation function when privacy is a concern. A common goal when choosing an allocation $\boldsymbol{f}$ is to minimise the variance of a particular statistic. That is, suppose that $\mathcal{C}_{\boldsymbol{f}}$ represents one-stage stratified sampling with allocation function $\boldsymbol{f}$. Then, given a population P and desired sample size m, the *optimal allocation function* $\boldsymbol{f}^*(P)$ with respect to a statistic θ is defined as

$$\boldsymbol{f}^*(P) = \arg\min_{\boldsymbol{f}} \operatorname{var}(\theta_{\mathcal{C}_{\boldsymbol{f}}}(P)), \tag{3}$$

where the randomness may come from both the allocation function and the sampling itself, and the minimum is over all allocation functions such that $\|\boldsymbol{f}(P)\|_1 \leq m$ for all P. [3]

[3] As an aside, we note that the notion of optimal allocations implicitly assumes that the historic or auxiliary data, H, used to inform the sampling design and the population data P are the same, or at

A natural question then is: what is the optimal allocation when one wants to compute the statistic of interest differentially privately? This is a simple yet subtle question. Our results in the previous sections indicate that the landscapes of optimal allocations in the non-private and private settings may be very different. This is a result of the fact that allocation functions that do not amplify well typically need to add more noise to achieve privacy (see discussion in Section 2.4). The additional noise needed to achieve privacy may overwhelm any gains in accuracy for the non-private statistic. Additionally, it is not immediately obvious how to define the optimal allocation in the private setting.

In this section, we formulate the notion of an optimal allocation under privacy constraints. Our goal is to initiate the study of alternative allocation functions that may prove useful when privacy is a concern. A full investigation of this question is outside the scope of this paper, but we provide some intuition for why this may be an interesting and important question for future work.

Given a statistic θ, we wish to define the optimal allocation for estimating θ privately. Let $\tilde{\theta}^\lambda$ be an λ-DP algorithm for estimating θ, so $\tilde{\theta}^\lambda(P)$ is an approximation of $\theta(P)$. The smaller λ is, the noisier $\tilde{\theta}^\lambda$ is. The scale of λ needed to ensure that $\tilde{\theta}^\lambda_{\mathcal{C}_{\boldsymbol{f}}}$ is ε-DP depends on the allocation function $\boldsymbol{f}$. Allocation functions that are very sensitive to changes in the input dataset will require more noise (smaller λ) to mask changes in the allocation. For any allocation $\boldsymbol{f}$, we will define the optimal parameter λ as that which minimises the maximum variance of $\tilde{\theta}^\lambda_{\mathcal{C}_{\boldsymbol{f}}}$ over all datasets P, while maintaining privacy:

$$\begin{aligned}\lambda_{\boldsymbol{f}} =& \arg\min_{\lambda>0} \sup_P \frac{\operatorname{var}(\tilde{\theta}^\lambda_{\mathcal{C}_{\boldsymbol{f}}}(P))}{\operatorname{var}(\theta_{\mathcal{C}_{\boldsymbol{f}}}(P))} \\ & \text{s.t. } \tilde{\theta}^\lambda_{\mathcal{C}_{\boldsymbol{f}}} \text{ is } (\varepsilon_1, \cdots, \varepsilon_k)\text{-stratified DP.}\end{aligned} \tag{4}$$

Now, by definition, $\tilde{\theta}^{\lambda_{\boldsymbol{f}}}_{\mathcal{C}_{\boldsymbol{f}}}$ is $(\varepsilon_1, \cdots, \varepsilon_k)$-stratified DP for any allocation function $\boldsymbol{f}$. We minimise the multiplicative increase in variance so that the supremum is not dominated by populations P for which $\operatorname{var}(\theta_{\mathcal{C}_{\boldsymbol{f}}}(P))$ is large. Given privacy parameters $\varepsilon_1, \cdots, \varepsilon_k \geq 0$, we now define the optimal allocation as the allocation function that minimises the maximum variance over all populations P:

$$\boldsymbol{f}^*_\varepsilon = \arg\min_{\boldsymbol{f}} \sup_P \operatorname{var}(\tilde{\theta}^{\lambda_{\boldsymbol{f}}}_{\mathcal{C}_{\boldsymbol{f}}}(P)). \tag{5}$$

where the minimum again is over all allocations $\boldsymbol{f}$ such that $\|\boldsymbol{f}(P)\|_1 \leq m$ for all P, and the supremum is over all populations of interest. This optimisation function has a different form to Eqn 3, which performs the optimisation independently for each population P. This difference is necessary in the private setting as we need to ensure that the choice of allocation function $\boldsymbol{f}^*_\varepsilon$ is not data dependent, since this would introduce additional privacy concerns. We can view the optimal allocation as the optimal balancing between the variance of the non-private statistic, and the scale of the noise needed to maintain privacy.

We believe that examining the difference between the optimal allocation in the non-private setting (Eqn (3)) and in the private setting (Eqn (5)) is an important question for future work. The main challenge is computing the parameter $\lambda_{\boldsymbol{f}}$ for every allocation $\boldsymbol{f}$. Analysing the privacy implications of $\boldsymbol{f}$ in the style of the previous sections gives us an upper bound on $\lambda_{\boldsymbol{f}}$, although this bound may be loose for specific statistics $\tilde{\theta}^\lambda$. So, while the previous sections developed our intuition for $\lambda_{\boldsymbol{f}}$, we believe new techniques are required to understand this parameter enough to solve Eqn (5).

least similar enough that $\boldsymbol{f}^*(H)$ is a good proxy for $\boldsymbol{f}^*(P)$. This provides further justification for the assumption that $H = P$ in our statements.

6.2 Challenges with optimal allocation

Optimal allocations are defined to perform well for a specific statistic of interest. However, in practice, a wide variety of analyses will be performed on the final sample. The chosen allocation function may be far from optimal for these other analyses. While this problem exists in the non-private setting, it becomes more acute in the private setting. An allocation function that is optimal for one statistic may result in privacy degradation (and hence low accuracy estimates) for another.

We illustrate this challenge using *Neyman allocation*, which is often employed for business surveys. Neyman allocation is the optimal allocation method for the weighted mean [26]:

$$\theta_\mu(S) = \frac{1}{|P|}\sum_{i=1}^{k}\frac{|P_i|}{|S_i|}\sum_{x\in S_i} x,$$

where $|P_i|$ is the size of stratum i, and $S_i = S \cap P_i$. The estimator $\theta_\mu(S)$ is an unbiased estimate of the population mean for any stratified sampling design. Given a desired sample size m, let $\boldsymbol{f}_{\text{Neyman}}$ be the allocation function corresponding to Neyman allocation. Provided each stratum is sufficiently large, $\boldsymbol{f}_{\text{Neyman}}(P) = (m_1, \cdots, m_k)$, where

$$m_i = \frac{|P_i|\sigma(P_i)}{\sum_{j=1}^{k}|P_j|\sigma(P_j)} \cdot m,$$

$\sigma^2(P_i)$ is the empirical variance in stratum i and sufficiently large means that $m_i \leq |P_i|$. Neyman allocation is deterministic and can be very sensitive to changes in the data due to its dependence on the variance within each stratum. So, while it can provide accurate results for some statistics, it provides very noisy results for other statistics of potential interest (e.g. privately computing strata sizes).

To demonstrate the sensitivity of Neyman allocation, we analysed the sensitivity on a real data set. The population is based on the County Business Patterns (CBP) data published by the U.S. Census Bureau [15].[4] Each data point is an establishment and the establishments are stratified by establishment size into $k = 12$ strata. With a target final sample size of $m = 10,000$, and using the weighted mean of the establishment size as the target statistic, the Neyman allocation for this population is [1261, 621, 517, 1969, 833, 1947, 1058, 762, 257, 248, 306, 225]. We can find a neighbouring population with Neyman allocation [1259, 620, 516, 1965, 831, 1943, 1056, 761, 257, 247, 306, 244]. While these allocations are not wildly different, they do differ by 19 samples in the top stratum, which might not have a large impact on the weighted mean, but could lead to more substantial changes for other statistics. As an illustrative example, we can consider the goal of privately estimating the stratum sizes in the sample, for which this allocation would lead to significant privacy degradation.

6.3 Privacy amplification from proportional sampling

Proportional sampling is an alternative allocation function that is used to provide equitable representation of each sub-population, or stratum. Given a desired sample size $m \in [n]$, proportional sampling samples an $r = \frac{m}{n}$ fraction of the data points (rounded to an integer)

[4] The data released by the U.S. Census Bureau is a tabulated version of the true micro data from the Business Register (BR), a database of all known single and multi-establishment employer companies. The data set we use is micro data generated to be consistent with the tabulated version. Each data point in this population is the size of an establishment in the US. In order to compute the sensitivity, we need to top code the data, we top code the data at 10,000.

from each stratum. Proportional sampling is not an optimal allocation in the non-private setting but, when implemented with randomised rounding, it has good privacy amplification. Now that we consider stratified sampling with number of stratums $k \geq 1$, we can state the following generalisation of Theorem 6.

▶ **Theorem 11** (Privacy Amplification for Proportional Sampling). *Let $r \in [0,1]$, $\varepsilon > 0$, $\mathcal{M}$ be an ε-DP mechanism, and $P = S_1 \sqcup \cdots \sqcup S_k$ and $P' = S'_1 \sqcup \cdots \sqcup S'_k$ be stratified neighboring datasets that differ on stratum i. If for all $j \in [k]$, $r|S_j| \geq 1$ and $r|S'_j| \geq 1$, then $\mathcal{M}_{C_{\boldsymbol{f}_{r,\text{prop}}}}$ is ε'-DP where*

$$\varepsilon' \leq \log\left(1 + 2r(e^{2\varepsilon} - 1)\right) + \log(1 + r(e^{2\varepsilon} - 1)).$$

Note that given a private statistic $\tilde{\theta}^{\lambda}$ as defined as above, this allows us to set $\lambda_{\boldsymbol{f}_{r,\text{prop}}} \approx \frac{\varepsilon}{6r}$, which is considerably larger than ε for small sampling rates. Thus, while proportional sampling may not minimise the variance of any single statistic, it may be a good choice since it performs reasonably well for *all* statistics.

7 Conclusion

In this paper, we have considered the privacy guarantees of sampling schemes, extending previous results to more complex and data-dependent sampling designs that are commonly used in practice. We find that considering these sampling schemes requires developing more nuanced analytical tools. In this work, we characterize the privacy impacts of randomized and data-dependent sampling schemes. Then, we apply our insights to analyze cluster and stratified sampling and to consider the question of optimal allocations under privacy. To the best of our knowledge, this work is the first to initiate study into these designs. As such, we hope to see future work in three areas. First, future work should tighten and optimize the constants in our theorems. Second, our results should be extended from pure to approximate (and other variants) of differential privacy. Finally, we hope to see further investigation into near-optimal allocations under privacy constraints.

References

1. Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016. doi:10.1145/2976749.2978318.
2. Julaiti Alafate and Yoav S Freund. Faster boosting with smaller memory. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/3ffebb08d23c609875d7177ee769a3e9-Paper.pdf.
3. Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *Journal of Privacy and Confidentiality*, 7, April 2015.
4. Hilal Asi and John C. Duchi. Near instance-optimality in differential privacy, 2020. arXiv:2005.10630.
5. Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 6280–6290, 2018.

6 Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy profiles and amplification by subsampling. *Journal of Privacy and Confidentiality*, 10(1), January 2020.

7 Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 464–473, Washington, DC, USA, 2014. IEEE Computer Society.

8 Johes Bater, Yongjoo Park, Xi He, Xiao Wang, and Jennie Rogers. SAQE: practical privacy-preserving approximate query processing for data federations. *Proc. VLDB Endow.*, 13(11):2691–2705, 2020. URL: `http://www.vldb.org/pvldb/vol13/p2691-bater.pdf`.

9 Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 437–454. Springer, 2010. `doi:10.1007/978-3-642-11799-2_26`.

10 Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 97–110. ACM, 2013. `doi:10.1145/2422436.2422450`.

11 Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 634–649, Washington, DC, USA, 2015. IEEE Computer Society.

12 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

13 Hamid Ebadi, Thibaud Antignac, and David Sands. Sampling and partitioning for differential privacy. In *14th Annual Conference on Privacy, Security and Trust, PST 2016, Auckland, New Zealand, December 12-14, 2016*, pages 664–673. IEEE, 2016. `doi:10.1109/PST.2016.7906954`.

14 Hamid Ebadi, David Sands, and Gerardo Schneider. Differential privacy: Now it's getting personal. *SIGPLAN Not.*, 50(1):69–81, January 2015.

15 Fabian Eckert, Teresa C. Fort, Peter K. Schott, and Natalie J. Yang. Imputing missing values in the us census bureau's county business patterns. Technical report, National Bureau of Economic Research, 2021.

16 Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3407–3416. PMLR, 2021. URL: `http://proceedings.mlr.press/v139/fraboni21a.html`.

17 Marco Gaboardi, Michael Hay, and Salil Vadhan. A programming framework for opendp. *Manuscript*, 2020.

18 Marco Gaboardi, James Honaker, Gary King, Jack Murtagh, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. Psi (ψ): A private data sharing interface. *arXiv preprint*, 2016. `arXiv:1609.04340`.

19 Antonious M. Girgis, Deepesh Data, Suhas N. Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2521–2529. PMLR, 2021. URL: `http://proceedings.mlr.press/v130/girgis21a.html`.

20 Jacob Imola and Kamalika Chaudhuri. Privacy amplification via bernoulli sampling. *arXiv preprint*, 2021. `arXiv:2105.10594`.

21 Joonas Jälkö, Antti Honkela, and Onur Dikmen. Differentially private variational inference for non-conjugate models. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL: `http://auai.org/uai2017/proceedings/papers/152.pdf`.

22 Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1023–1034, 2015.

23 Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

24 Wouter Kool, Herke van Hoof, and Max Welling. Estimating gradients for discrete random variables by sampling without replacement. In *International Conference on Learning Representations*, 2020. URL: `https://openreview.net/forum?id=rklEj2EFvB`.

25 Levi H. S. Lelis, Roni Stern, Shahab Jabbari Arfaee, Sandra Zilles, Ariel Felner, and Robert C. Holte. Predicting optimal solution costs with bidirectional stratified sampling in regular search spaces. *Artif. Intell.*, 230:51–73, 2016. `doi:10.1016/j.artint.2015.09.012`.

26 Jerzy Neyman. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–606, 1934.

27 Hieu Tat Nguyen and Arnold W. M. Smeulders. Active learning using pre-clustering. In Carla E. Brodley, editor, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004. `doi:10.1145/1015330.1015349`.

28 Mijung Park, James R. Foulds, Kamalika Chaudhuri, and Max Welling. Variational bayes in private settings (VIPS). *J. Artif. Intell. Res.*, 68:109–157, 2020. `doi:10.1613/jair.1.11763`.

29 Adam Smith. Differential privacy and the secrecy of the sample, February 2010. URL: `https://adamdsmith.wordpress.com/2009/09/02/sample-secrecy/`.

30 Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter 7, pages 347–450. Springer International Publishing AG, Cham, Switzerland, 2017.

31 Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled renyi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1226–1235. PMLR, 2019. URL: `http://proceedings.mlr.press/v89/wang19b.html`.

32 Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2493–2502. JMLR.org, 2015. URL: `http://proceedings.mlr.press/v37/wangg15.html`.

33 Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle. *J. Mach. Learn. Res.*, 17:183:1–183:40, 2016.

34 Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. A minimax theory for adaptive data analysis. *arXiv preprint*, 2016. `arXiv:1602.04287`.

35 Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

A Basic facts about indistinguishability

▶ **Definition 12.** *Let the* LCS distance *between two data sets P and P', denoted $d_{LCS}(P, P')$, be the minimal k such that if we let $P = P_0$ and $P' = P_k$, there exist data sets $P_1, P_2, \cdots, P_{k-1}$ where for all $i = 0, \cdots, k-1$, P_i and P_{i+1} are unbounded neighbors.*

▶ **Lemma 13** ([12])**.** *Let X, Y and Z be random variables. For any $\varepsilon, \varepsilon' > 0$, if X and Y are ε-indistinguishable, and Y and Z are ε'-indistinguishable, then X and Z are $\varepsilon + \varepsilon'$-indistinguishable.*

Many of our proofs use couplings so let us briefly describe on the main method we will use to construct a coupling of two random variables. Let X be a random variable taking values in Ω_X and Y be a random variable taking values in Ω_Y. Suppose there exists a (possibly randomised) transformation $f : \Omega_X \to \Omega_Y$ such that $Y = f(X)$. That is, for all $y \in \Omega_Y$, $\Pr(Y = y) = \sum_{x \in \Omega_X} \Pr(X = x) \Pr(f(x) = y)$. Then we can construct a coupling of X and Y by $\mu(x, y) = \Pr(X = x) \Pr(f(x) = y)$. A short calculation confirms that this defines a coupling. Further, notice that $\mu(x, y) \neq 0$ if and only $\Pr(f(x) = y) \neq 0$.

▶ **Lemma 14.** *Let X and Y be random variables taking values in $\mathcal{U}^*$ such that there exists a coupling μ such that if $\mu(x, y) \neq 0$ then the LCS distance between x and y is at most A. Then if $\mathcal{M}$ is ε-unbounded DP then $\mathcal{M}(X)$ and $\mathcal{M}(Y)$ are $A\varepsilon$-indistinguishable.*

▶ **Lemma 15** (Advanced joint convexity, [6])**.** *Let X and Y be random variables satisfying $X = (1-q)X_0 + qX_1$ and $Y = (1-q)Y_0 + qY_1$ for some $q \in [0, 1]$ and random variables X_0, X_1, Y_0 and Y_1. If X_0 and Y_0 are ε-indistinguishable, X_1 and Y_1 are $\varepsilon+\varepsilon'$-indistinguishable, X_1 and Y_0 are $\varepsilon + \varepsilon'$-indistinguishable, and X_0 and Y_1 are $\varepsilon + \varepsilon'$-indistinguishable, then X and Y are $\varepsilon + \log(1 + q(e^{\varepsilon'} - 1))$-indistinguishable.*

B Randomized data-independent sampling

▶ **Lemma 16.** *Given $m \in \mathbb{N}$, define $\mathcal{C}_m : \mathcal{U}^* \to \mathcal{U}^m$ be defined as follows: given a dataset P, form a sample S by sampling m data points randomly without replacement from P, then $\mathcal{C}_m(P) = S$. Let P and P' be unbounded neighboring datasets and $m, m' \in \mathbb{N}$, then $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_{m'}}(P')$ are*

$$\left(\log\left(1 + \frac{m}{|P|+1}(e^{2\varepsilon} - 1)\right) + |m - m'|\varepsilon\right) \text{ - indistinguishable.}$$

Proof. Let $P' = P \cup \{x\}$. First, let us focus on the case where $m' = m$. Now,

$$\begin{aligned}\mathcal{M}_{\mathcal{C}_m}(P') &= \frac{\binom{|P|}{m}}{\binom{|P|+1}{m}}\mathcal{M}_{\mathcal{C}_m}(P) + \left(1 - \frac{\binom{|P|}{m}}{\binom{|P|+1}{m}}\right)\mathcal{M}(\mathcal{C}_m(P')|_{x\in S}) \\ &= \left(1 - \frac{m}{|P|+1}\right)\mathcal{M}_{\mathcal{C}_m}(P) + \frac{m}{|P|+1}\mathcal{M}(\mathcal{C}_m(P')|_{x\in S}),\end{aligned}$$

where $\mathcal{C}_m(P')|_{x\in S}$ denotes the random variable $\mathcal{C}_m(P')$ conditioned on the event that $x \in S$. Now, we can define a coupling of $\mathcal{C}_m(P)$ and $\mathcal{C}_m(P')|_{x\in S}$ by first sampling S from $\mathcal{C}_m(P)$, then replacing a random element of S by x. This coupling has LCS distance at most 2, so by Lemma 14, $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}(\mathcal{C}_m(P')|_{x\in S}))$ are 2ε-indistinguishable. Thus, by Lemma 15, $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_m}(P')$ are $\log\left(1 + \frac{m}{|P|+1}(e^{2\varepsilon} - 1)\right)$ -indistinguishable.

Next, let us consider the case $|m - m'| = 1$ and $P = P'$. We can define a coupling of $\mathcal{C}_m(P)$ and $\mathcal{C}_{m'}(P)$ as follows: first sample S from $\mathcal{C}_m(P)$, then add a random element of $P \backslash S$ to S. This coupling has LCS distance at most 1, so by Lemma 14, $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_{m'}}(P)$ are ε-indistinguishable.

Finally, we'll use Lemma 13 to complete the proof. Note that $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_m}(P')$ are $\log\left(1 + \frac{m}{|P|+1}(e^{2\varepsilon} - 1)\right)$-indistinguishable. Then there exist $m_1, \cdots, m_{\ell-1}$ such that if we set $m_0 = m$ and $m_{|m-m'|} = m'$ then for all i, $|m_i - m_{i-1}| \leq 1$ and so $\mathcal{M}(\mathcal{C}_{m_{i-1}}(P'))$ and $\mathcal{M}(\mathcal{C}_{m_i}(P'))$ are ε-indistinguishable. Therefore, by Lemma 13, $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_{m'}}(P')$ are $\left(\log\left(1 + \frac{m}{|P|+1}(e^{2\varepsilon} - 1)\right) + |m - m'|\varepsilon\right)$ - indistinguishable. ◀

▶ **Definition 17** (log-Lipschitz functions). *A function $q : [n] \to \mathbb{R}_{\geq 0}$ is* ε-log-Lipschitz *if for all $m \in \{0, 1, \ldots, n-1\}$, $|\log q(m) - \log q(m+1)| \leq \varepsilon$.*

▶ **Lemma 18.** *Let $w : [n] \to \mathbb{R}_{\geq 0}$ be nondecreasing, and let $p : [n] \to \mathbb{R}_{\geq 0}$ be any function. Then,*

$$\max_{q:[n]\to\mathbb{R}_{\geq 0} \text{ is } \varepsilon\text{-log-Lipschitz}} \frac{\sum_{m=0}^{n} q(m)w(m)p(m)}{\sum_{m=0}^{n} q(m)p(m)} \leq \frac{\sum_{m=0}^{n} e^{\varepsilon m}w(m)p(m)}{\sum_{m=0}^{n} e^{\varepsilon m}p(m)}$$

The proof of Lemma 18 is omitted due to space constraints.

Proof of Theorem 3. Let $\mathcal{C}_m : \mathcal{U}^* \to \mathcal{U}^m$ be the sampling scheme that given a dataset P, returns S where S is a uniformly random subset of P of size m (drawn without replacement). Let $y \in \mathcal{O}$ be any outcome, and let $P \sim P'$ be neighboring datasets. Then, we have that

$$\begin{aligned}
\Pr[\mathcal{M}_{\mathcal{C}}(P) = y] &= \sum_{m=0}^{n} \Pr[\mathcal{M}_{\mathcal{C}_m}(P) = y] \cdot \Pr[|\mathcal{C}(P)| = m] \\
&\leq \sum_{m=0}^{n} \left(1 + \frac{m}{n}(e^{\varepsilon} - 1)\right) \cdot Pr[\mathcal{M}_{\mathcal{C}_m}(P') = y] \cdot t(m) \\
&\leq \frac{\sum_{m=0}^{n} \left(1 + \frac{m}{n}(e^{\varepsilon} - 1)\right) \cdot e^{\varepsilon m} \cdot t(m)}{\sum_{m=0}^{n} e^{\varepsilon m} t(m)} \cdot \sum_{m=0}^{n} \Pr[\mathcal{M}_{\mathcal{C}_m}(P') = y] \cdot t(m) \\
&= \left(1 + \frac{\mathbb{E}_{m\sim\tilde{t}}[m]}{n}(e^{\varepsilon} - 1)\right) \cdot \Pr[\mathcal{M}_{\mathcal{C}}(P') = y]
\end{aligned}$$

where the first inequality follows from Lemma 16. Then, note that $(1 + (m/n)(e^{\varepsilon} - 1))$ is non-decreasing, and that $\Pr[\mathcal{M}_{\mathcal{C}_m}(P')) = y]$ is ε-log-Lipschitz by definition, so the second inequality follows by Lemma 18. After rearranging and simplifying, we obtain the desired result.

Finally, for the lower bound, suppose the data universe $\mathcal{U} = [0, 1]$. Let $P = \{1, \cdots, 1\}$ consist of n 1s and P' be the neighboring dataset $P' = P \backslash \{1\} \cup \{0\}$. Let $\mathcal{M} : \mathcal{U}^* \to \mathbb{R}$ be defined by $\mathcal{M}(S) = \sum_{x \in S} \mathbb{1}\{x = 1\} + \mathrm{Lap}\,(1/\varepsilon)$ so $\mathcal{M}$ is ε-unbounded DP. Then

$$\begin{aligned}
\frac{\Pr(\mathcal{M}_{\mathcal{C}}(P') = n)}{\Pr(\mathcal{M}_{\mathcal{C}}(P) = n)} &= \frac{\sum_{m=0}^{n} \Pr(t = m)\left(\frac{m}{n}e^{-(n-m+1)\varepsilon} + \left(1 - \frac{m}{n}\right)e^{-(n-m)\varepsilon}\right)}{\sum_{m=0}^{n} \Pr(t = m)e^{-(n-m)\varepsilon}} \\
&= 1 - \frac{1}{n}(1 - e^{-\varepsilon})\frac{\sum_{m=0}^{n} \Pr(t = m)e^{m\varepsilon}m}{\sum_{m=0}^{n} \Pr(t = m)e^{m\varepsilon}}.
\end{aligned}$$

Thus, taking the reciprocal,

$$\log \frac{\Pr(\mathcal{M}_{\mathcal{C}}(P) = n)}{\Pr(\mathcal{M}_{\mathcal{C}}(P') = n)} = -\log\left(1 - \frac{1}{n}(1 - e^{-\varepsilon})\frac{\sum_{m=0}^{n} \Pr(t = m)e^{m\varepsilon}m}{\sum_{m=0}^{n} \Pr(t = m)e^{m\varepsilon}}\right). \quad ◀$$

C Data-dependent sampling

Proof of Proposition 4: hypothesis testing perspective. Let $H : \mathbb{N} \to \{0,1\}$ be the hypothesis test such that for all $x \in \mathbb{N}$, and $b \in \{0,1\}$, $e^{-\varepsilon} \Pr(H(x) = b) \leq \Pr(H(x+1) = b) \leq e^{\varepsilon} \Pr(H(x) = b)$. Then $H' : \mathcal{U}^* \to \{0,1\}$ defined by $H'(S) = H(|S|)$ is ε-unbounded DP. By assumption, $H'_{\mathcal{C}_{\tilde{f}}}$ is ε'-DP. This implies that $H(\tilde{f}(P))$ and $H(\tilde{f}(P'))$ are ε'-indistinguishable. Therefore,

$$\text{adv}(H) = \Pr[H(\tilde{f}(P)) = 0] - \Pr[H(\tilde{f}(P')) = 0] \leq \Pr[H(\tilde{f}(P')) = 0](e^{\varepsilon'} - 1) \leq e^{\varepsilon'} - 1.$$

The result follows from taking the supremum over all ε-DP H. ◀

Proof of Proposition 5. Define $\mathcal{M}_{SS} : \mathcal{U}^* \to \mathbb{N}$ as follows. For all $P \in \mathcal{U}^*$, $\mathcal{M}(P) = |P| + \text{Lap}(1/\varepsilon)$. Then $\mathcal{M}$ is ε-unbounded DP. Suppose that $\tilde{f} : \mathcal{U}^* \to \mathbb{N}$ is such that for all ε-unbounded DP mechanisms $\mathcal{A}$, $\mathcal{A}_{\mathcal{C}_{\tilde{f}}}$ is ε'-bounded DP. This implies that $\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) = \tilde{f}(P) + \text{Lap}(1/\varepsilon)$ is ε'-bounded DP. Therefore, by the definition of α, there exists a population P such that $\sup_{P \in \mathcal{U}^n} \mathbb{E}[|\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) - f(P)|^2] \geq \alpha$. Also

$$\alpha \leq \mathbb{E}[|\mathcal{M}_{\mathcal{C}_{\tilde{f}}}(P)) - f(P)|^2] = \mathbb{E}[|\tilde{f}(P) + \text{Lap}(1/\varepsilon) - f(P)|^2] = \mathbb{E}[]|\tilde{f}(P) - f(P)|^2] + (1/\varepsilon)^2.$$

After a small amount of rearranging we arrive at the result. ◀

Proof of Theorem 6: proportional allocation with randomized rounding. Let P be a dataset, x be a data point and $P' = P \cup \{x\}$. Let $m = r|P|$, $m' = r|P'|$, $m^L = \lfloor m \rfloor$, $m'^L = \lfloor m' \rfloor$, $p = m - m^L$ and $p' = m' - m'^L$. Now, $m' - m = r < 1$ so we have two cases, $m^L = m'^L$ or $m^L = m'^L - 1$.

As in Lemma 16, let $\mathcal{C}_m : \mathcal{U}^* \to \mathcal{U}^m$ be the sampling scheme that given a dataset P, returns S where S is a uniformly random subset of P of size m (drawn without replacement). Note that by Theorem 2, for $m, m' \in \mathbb{N}$, $\mathcal{M}_m(P)$ and $\mathcal{M}_{m'}(P)$ are $|m-m'|\varepsilon$-indistinguishable, and $\mathcal{M}_{\mathcal{C}_m}(P)$ and $\mathcal{M}_{\mathcal{C}_{m'}}(P')$ are $\log\left(1 + \frac{m}{|P|+1}(e^{2\varepsilon} - 1)\right) + |m - m'|\varepsilon$-indistinguishable.

Firstly, suppose $m^L = m'^L$. Let $\mu_0 = \frac{1}{1-r}((1-p-r)\mathcal{M}_{\mathcal{C}_{m^L}}(P) + p\mathcal{M}_{\mathcal{C}_{m^L+1}}(P))$, $\mu'_0 = \frac{1}{1-r}((1-p-r)\mathcal{M}_{\mathcal{C}_{m^L}}(P') + p\mathcal{M}_{\mathcal{C}_{m^L+1}}(P'))$, $\mu_1 = \mathcal{M}_{\mathcal{C}_{m^L}}(P)$, and $\mu'_1 = \mathcal{M}_{\mathcal{C}_{m^L+1}}(P')$.

Notice that $\mathcal{M}_{\mathcal{C}_r}(P) = (1-r)\mu_0 + r\mu_1$ and $\mathcal{M}_{\mathcal{C}_r}(P') = (1-r)\mu'_0 + r\mu'_1$. Now, by Lemma 15 and Lemma 14, μ_0 and μ'_0 are $\log(1 + \frac{m^L+1}{|P|+1}(e^{2\varepsilon} - 1))$-indistinguishable. Further, all the pairs (μ'_0, μ_1), (μ_1, μ'_1) and (μ_0, μ'_1) are $\left(\log(1 + \frac{m^L+1}{|P|+1}(e^{2\varepsilon} - 1)) + \varepsilon\right)$-indistinguishable. Therefore, by Lemma 15, $\mathcal{M}_{\mathcal{C}_r}(P)$ and $\mathcal{M}_{\mathcal{C}_r}(P')$ are ε'-indistinguishable where $\varepsilon' \leq \log\left(1 + \frac{m^L+1}{|P|+1}(e^{2\varepsilon} - 1)\right) + \log(1 + r(e^{\varepsilon} - 1)) \leq \log\left(1 + \left(r + \frac{1}{|P|+1}\right)(e^{2\varepsilon} - 1)\right) + \log(1 + r(e^{\varepsilon} - 1))$.

Next, suppose $m'^L = m^L + 1$. Let $1 - q = \min\{p, 1 - p'\}$ and $\mu_0 = \mathcal{M}_{\mathcal{C}_{m^L+1}}(P)$, $\mu'_0 = \mathcal{M}_{\mathcal{C}_{m^L+1}}(P')$, $\mu_1 = \frac{1}{q}((p-1+q)\mathcal{M}_{\mathcal{C}_{m^L+1}}(P) + (1-p)\mathcal{M}_{\mathcal{C}_{m^L}}(P),)$, and $\mu'_1 = \frac{1}{q}((1-p'-1+q)\mathcal{M}_{\mathcal{C}_{m^L+1}}(P') + p'\mathcal{M}_{\mathcal{C}_{m^L+2}}(P'))$. Notice that $\mathcal{M}_{\mathcal{C}_r}(P) = (1-q)\mu_0 + q\mu_1$ and $\mathcal{M}_{\mathcal{C}_r}(P') = (1-q)\mu'_0 + q\mu'_1$. Now, by Lemma 2, μ_0 and μ'_0 are $\log\left(1 + \frac{m^L+1}{|P|+1}\right)$-indistinguishable. Further, all the pairs (μ'_0, μ_1), (μ_1, μ'_1) and (μ_0, μ'_1) are $\left(\log(1 + \frac{m^L+1}{|P|+1}(e^{2\varepsilon} - 1)) + 2\varepsilon\right)$-indistinguishable. Also, note that $q \leq r$. Then by Lemma 15, $\mathcal{M}_{\mathcal{C}_r}(P)$ and $\mathcal{M}_{\mathcal{C}_r}(P')$ are ε'-indistinguishable where $\varepsilon' \leq \log\left(1 + \frac{m^L+1}{|P|+1}(e^{2\varepsilon} - 1)\right) + \log(1 + p(e^{2\varepsilon} - 1)) \leq \log\left(1 + \left(r + \frac{1}{|P|+1}\right)(e^{2\varepsilon} - 1)\right) + \log(1 + r(e^{2\varepsilon} - 1))$. ◀

D Cluster sampling

Proof of Theorem 8. Without loss of generality, let $i = 1$. Notice that conditioned on cluster $1 \notin I$, the distribution of outputs of $\mathcal{M}_{\mathcal{C}}(P)$ and $\mathcal{M}_{\mathcal{C}}(P')$ are identical. Let E be a set of outcomes. Then

$$\begin{aligned}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E) &= \frac{\ell}{k}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E \mid 1 \in I) + \left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E \mid 1 \notin I) \\ &= \frac{\ell}{k}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E \mid 1 \in I) + \left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \notin I).\end{aligned}$$

Now, we have that $\frac{\ell}{k}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E \mid 1 \in I) = \frac{\ell}{k}\sum_{|I|=\ell, 1\in I}\frac{1}{\binom{k}{\ell}}\Pr(\mathcal{M}(P_I) \in E) \leq \frac{\ell}{k}\sum_{|I|=\ell,1\in I}\frac{1}{\binom{k}{\ell}}e^{\varepsilon}\Pr(\mathcal{M}(P'_I) \in E) = \frac{\ell}{k}e^{\varepsilon}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I)$, where the inequality follows from the fact that the LCS distance between P_I and P'_I is 1. Thus,

$$\begin{aligned}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E) &\leq \frac{\ell}{k}e^{\varepsilon}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I) + \left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \notin I) \\ &= \Pr(\mathcal{M}_{\mathcal{C}}(P') \in E) + \frac{\ell}{k}(e^{\varepsilon} - 1)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I).\end{aligned}$$

Now, we need to relate $\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I)$ to $\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E)$. For a set I such that $1 \notin I$ and index $i \in I$, let $I \cup \{1\}\backslash\{i\}$ be the set where index i has been replaced with 1. Then,

$$\begin{aligned}\left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \notin I) &= \sum_{|I|=\ell,1\notin I}\frac{1}{\binom{k}{\ell}}\Pr(\mathcal{M}(P_I) \in E) \\ &= \sum_{|I|=\ell,1\notin I}\sum_{i\in I}\frac{1}{\ell}\frac{1}{\binom{k}{\ell}}\Pr(\mathcal{M}(P_I) \in E) \\ &\geq \sum_{|I|=\ell,1\notin I}\sum_{i\in I}\frac{1}{\ell}\frac{1}{\binom{k}{\ell}}e^{-(n_1+n_i)\varepsilon}\Pr(\mathcal{M}(P_{I\cup\{1\}\backslash\{i\}}) \in E) \\ &\geq e^{-(n_1+n_{\max})\varepsilon}\frac{1}{\ell}\sum_{|I|=\ell,1\notin I}\sum_{i\in I}\frac{1}{\binom{k}{\ell}}\Pr(\mathcal{M}(P_{I\cup\{1\}\backslash\{i\}}) \in E),\end{aligned}$$

where the first inequality follows from the fact that the LCS distance between P_I and $P_{I\cup\{1\}\backslash\{i\}}$ is at most $n_1 + n_i$. Now, notice that the sets $I \cup \{1\}\backslash\{i\}$ in the above sum all contain 1, and each index I' such that $|I'| = \ell$ and $1 \in I'$ appears in the sum $k - \ell$ times (corresponding to the $k - \ell$ possible choices for the swapped index i). Therefore, we can rewrite the sum as

$$\begin{aligned}\left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \notin I) &\geq e^{-(n_1+n_{\max})\varepsilon}\frac{k-\ell}{\ell}\sum_{|I|=\ell,1\in I}\frac{1}{\binom{k}{\ell}}\Pr(\mathcal{M}(P_I) \in E) \\ &= e^{-(n_1+n_{\max})\varepsilon}\left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I).\end{aligned}$$

Thus, we can complete the proof with the following steps.

$$\begin{aligned}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E) &= \frac{\ell}{k}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I) + \left(1 - \frac{\ell}{k}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \notin I) \\ &\geq \frac{\ell}{k}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I) + \left(1 - \frac{\ell}{k}\right)e^{-(n_1+n_{\max})\varepsilon}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I) \\ &= \left(\frac{\ell}{k} + \left(1 - \frac{\ell}{k}\right)e^{-(n_1+n_{\max})\varepsilon}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I).\end{aligned}$$

$$\begin{aligned}\Pr(\mathcal{M}_{\mathcal{C}}(P) \in E) &\le \Pr(\mathcal{M}_{\mathcal{C}}(P') \in E) + \frac{\ell}{k}(e^{\varepsilon}-1)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E \mid 1 \in I) \\ &\le \Pr(\mathcal{M}_{\mathcal{C}}(P') \in E) + \frac{\ell}{k}(e^{\varepsilon}-1)\frac{1}{\left(\frac{\ell}{k} + \left(1-\frac{\ell}{k}\right)e^{-(n_1+n_{\max})\varepsilon}\right)}\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E) \\ &\le \left(1 + \frac{\ell}{k}(e^{\varepsilon}-1)\frac{1}{\left(\frac{\ell}{k} + \left(1-\frac{\ell}{k}\right)e^{-(n_1+n_{\max})\varepsilon}\right)}\right)\Pr(\mathcal{M}_{\mathcal{C}}(P') \in E)\end{aligned}$$

◀

Proof of Theorem 7. Let $C_1 = \{1, \cdots, 1\}$ and $C_j = \{-1, \cdots, -1\}$ for all $j \in \{2, \cdots, k\}$. Let $C_1' = C_1 \backslash \{1\} \cup \{-1\}$ be the same as C_1 except with one 1 switched to a -1. Let $\mathcal{M}(S) = \sum_{x \in S} x + \text{Lap}(1/\varepsilon)$, so $\mathcal{M}$ is ε-unbounded DP. Notice that $\mathcal{M}$ has the property that if $\sum_{x \in S'} x = \sum_{x \in S} x + a$, for some $a \in \mathbb{R}$ then $\Pr(\mathcal{M}(S) = \sum_{x \in S} x) = e^{|a|\varepsilon}\Pr(\mathcal{M}(S') = \sum_{x \in S} x)$. This equality allows us to tighten many of the inequalities that appeared in the proof of Theorem 8 and give a lower bound. We omit the rest of the proof due to space constraints. ◀

Proof of Theorem 9. Let $D = \mathcal{C}_\ell(P)$ and $D' = \mathcal{C}_\ell(P')$. For an event $E \in \mathcal{O}$, define the probabilities p, q, p' and q' as follows.

$$\begin{aligned} p &= \Pr(\mathcal{M}(D) \in E | C_1 \in D) & q &= \Pr(\mathcal{M}(D) \in E | C_1 \notin D) \\ p' &= \Pr(\mathcal{M}(D') \in E | C_1 \in D') & q' &= \Pr(\mathcal{M}(D') \in E | C_1 \notin D')\end{aligned}$$

By the existence of $\mathcal{H}$ described in the lemma statement, there must exist an event E such that $q \le e^{-\varepsilon'}p$. Since P and P' only differ on C_1, the distributions of $\mathcal{M}(D)|_{C_1 \notin D}$ and $\mathcal{M}(D')|_{C_1 \notin D'}$ are identical, which means that $q = q'$. Then, we can compute a lower bound on the indistinguishability of $\mathcal{M}(D)$ and $\mathcal{M}(D')$ as follows. Without loss of generality, assume $p' > p$, and proceed as follows.

$$\begin{aligned}\frac{\Pr(\mathcal{M}(D') \in E)}{\Pr(\mathcal{M}(D) \in E)} &= \frac{p' \cdot Pr(C_1 \in D') + q \cdot Pr(C_1 \notin D')}{p \cdot Pr(C_1 \in D) + q \cdot Pr(C_1 \notin D)} = \frac{p' \cdot \frac{\ell}{k} + q \cdot (1 - \frac{\ell}{k})}{p \cdot \frac{\ell}{k} + q \cdot (1 - \frac{\ell}{k})} \\ &\ge 1 + \frac{(p'-p)\frac{\ell}{k}}{p \cdot (\frac{\ell}{k} + e^{-\varepsilon'}(1 - \frac{\ell}{k}))} = 1 + \left(\frac{p'}{p} - 1\right)\frac{\frac{\ell}{k}}{\frac{\ell}{k} + e^{-\varepsilon'}(1-\frac{\ell}{k})}\end{aligned}$$

where the final inequality follows from the fact that $\mathcal{M}$ is ϵ-DP, so $p'/p \ge e^{\epsilon''}$ by definition. ◀

E Stratified sampling

Proof of Theorem 11: proportional allocation for stratified sampling. Given $\mathcal{M} : ([k] \times \mathcal{U})^* \to \mathcal{Y}$, for all datasets $T_2, \cdots, T_k \in \mathcal{U}^*$, define $\mathcal{M}^{T_2,\cdots,T_k} : \mathcal{U}^* \to \mathcal{Y}$ by $\mathcal{M}^{T_2,\cdots,T_k}(S) = \mathcal{M}(S \sqcup T_2 \sqcup \cdots \sqcup T_k)$. Then since $\mathcal{M}$ was $(\varepsilon, \cdots, \varepsilon)$-stratified unbounded DP, $\mathcal{M}^{T_2,\cdots,T_k}$ is ε-unbounded DP. Let $\mathcal{C}_r$ be as in Lemma 6 so for all S, S' unbounded neighbours such that $r|S| \ge 1$ and $r|S'| \ge 1$, $\mathcal{M}_{\mathcal{C}_r}^{T_2,\cdots,T_k}(S)$ and $\mathcal{M}_{\mathcal{C}_r}^{T_2,\cdots,T_k}(S')$ are ε'-indistinguishable where $\varepsilon' \le \log\left(1 + 2r(e^{2\varepsilon} - 1)\right) + \log(1 + r(e^{2\varepsilon} - 1))$. Now, let $P = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_k$ and $P = S_1' \sqcup S_2 \sqcup \cdots \sqcup S_k$ be unbounded stratified neighboring datasets that differ in the first stratum. Since $S_2 \sqcup \cdots \sqcup S_k$ are shared between P and P', and the datasets T_i only dependent on strata S_i, the distribution of $T_2, \cdots, T_k$ are identical given inputs P and P'. Let q be the distribution of $T_2, \cdots, T_k$ so $q(T_2, \cdots, T_k) = \Pr(\mathcal{C}_r(S_2) = T_2, \cdots, \mathcal{C}_r(S_k) = T_k)$. Then given an event E, $\Pr(\mathcal{M}_{\mathcal{C}_{f_{\text{prop},r}}}(P) \in E) = \int_{T_2,\cdots,T_k} q(T_2, \cdots, T_k)\Pr(\mathcal{M}_{\mathcal{C}_r}^{T_2,\cdots,T_k}(S_1) \in E) \le \int_{T_2,\cdots,T_k} q(T_2, \cdots, T_k)e^{\varepsilon'}\Pr(\mathcal{M}_{\mathcal{C}_r}^{T_2,\cdots,T_k}(S_1') \in E) = e^{\varepsilon'}\Pr(\mathcal{M}_{\mathcal{C}_{f_{\text{prop},r}}}(P') \in E)$. ◀

Leximax Approximations and Representative Cohort Selection

Monika Henzinger
Universität Wien, Austria

Charlotte Peale
Stanford University, CA, USA

Omer Reingold
Stanford University, CA, USA

Judy Hanwen Shen
Stanford University, CA, USA

Abstract

Finding a representative cohort from a broad pool of candidates is a goal that arises in many contexts such as choosing governing committees and consumer panels. While there are many ways to define the degree to which a cohort represents a population, a very appealing solution concept is lexicographic maximality (leximax) which offers a natural (pareto-optimal like) interpretation that the utility of no population can be increased without decreasing the utility of a population that is already worse off. However, finding a leximax solution can be highly dependent on small variations in the utility of certain groups. In this work, we explore new notions of approximate leximax solutions with three distinct motivations: better algorithmic efficiency, exploiting significant utility improvements, and robustness to noise. Among other definitional contributions, we give a new notion of an approximate leximax that satisfies a similarly appealing semantic interpretation and relate it to algorithmically-feasible approximate leximax notions. When group utilities are linear over cohort candidates, we give an efficient polynomial-time algorithm for finding a leximax distribution over cohort candidates in the exact as well as in the approximate setting. Furthermore, we show that finding an integer solution to leximax cohort selection with linear utilities is NP-Hard.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains

Keywords and phrases fairness, cohort selection, leximin, maxmin

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.2

Related Version *Full Version*: https://arxiv.org/abs/2205.01157

Funding *Monika Henzinger*: This work was done in part as Stanford University Distinguished Visiting Austrian Chair. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019564 "The Design of Modern Fully Dynamic Data Structures (MoDynStruct)" and from the Austrian Science Fund (FWF) project "Fast Algorithms for a Reactive Network Layer (ReactNet)", P 33775-N, with additional funding from the *netidee SCIENCE Stiftung*, 2020–2024.
Charlotte Peale: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.
Omer Reingold: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness, the Sloan Foundation Grant 2020-13941 and the Simons Foundation investigators award 689988.
Judy Hanwen Shen: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

1 Introduction

In many fairness-related settings, we seek to select an outcome that does not disproportionately harm any key subgroup. Speaking in terms of group utilities, a fair solution would ideally provide every key subgroup with high utility. Unfortunately, such a goal may be impossible to achieve if the utilities derived by subgroups from any potential solutions are in opposition. Moreover, other goals such as seeking to equalize utilities across groups may artificially constrain the utility of certain groups in order to match some group with uniformly low utility.

The classic *maximin* objective, which seeks to output solutions that maximize the utility of the worst-off group, has been widely studied as a goal that can circumvent these potential pitfalls by seeking to achieve the best possible outcome for the worst-off group. This results in a set of solutions that optimize the outcome for the worst-off group, but may still vary quite a bit with respect to the second-worst-off group, third-worst-off group, etc. *Lexicographically maximal* solutions strengthen the maximin objective by requiring that the utility of the second-worst-off group be maximized subject to the worst-off-group achieving its maximin value, the third-worst-off group be maximized subject to the worst-off and second-worst-off values, and so on. This goal intuitively tells us that a lexicographically maximal solution gives the best-possible utility guarantee we can give for each group without harming another group.

Lexicographic maximality (which we refer to as *leximax*, but is sometimes referred to in the literature as *leximin*) has been widely studied in the context of allocations [14, 16, 19]. Recently, Diana, Gill, Globus-Harris, Kearns, Roth, and Sharifi-Malvajerdi [10] explored applying the objective to the contemporary fairness context of loss minimization. In this paper, motivated by the goal of selecting a representative cohort from a group of candidates, we generalize the approach of [10] to the goal of selecting a solution that achieves lexicographically maximal utilities for a set of key subgroups.

Our contributions fall into two main categories: definitional, where we explore useful variants of the leximax objective and their relations in the general setting of selecting a leximax solution from a set of potential solutions, and algorithmic, in which we investigate how to efficiently find exact leximax solutions as well as different variants in the specific context of selecting representative cohorts. We provide an overview of definitional contributions in Section 1.1, followed by an overview of the cohort selection context and resulting algorithms in Section 1.2.

1.1 Approximations of Lexicographically Maximal Solutions

Diana et al. [10] define an approximate notion of lexicographic maximality for which they construct oracle-efficient algorithms. Their notion is influenced by an algorithmic approach to calculating leximax solutions in that it assumes the maximal values of the worst-off group, second-worst-off group, etc. are calculated recursively based on whatever estimates came before. The definition assumes some small amount of error when calculating the maximin utility value, and then considers how this error would propagate to the second-worst-off-group's maximum value, then considers how additional errors around the second-worst-off-group's maximum value together with errors from the worst-off group maximum value might propagate to the third-worst-off group, and so on.

One of the appealing aspects of leximax solutions is that they offer a simple semantic interpretation that explains the sort of fairness guarantees such solutions provide: given a leximax solution, any alternative solution that improves the utility of some group must also decrease the utility of some worse-off group (Proposition 4). While the approximation

notion presented by [10] is very natural, they also show that such approximate solutions may greatly diverge from exact solutions (see Example 5 for details), meaning that they may also diverge from this appealing semantic interpretation.

Ideally, we'd like a well-defined notion of approximation that extends the semantic interpretation of leximax and relates to the algorithmically achievable notion presented by [10]. However, we find that such a definition is somewhat difficult to pin down. *Many natural relaxations of the semantic definition result in notions of approximations where either no solutions are guaranteed to satisfy the notion or the notions themselves may not imply a meaningful fairness guarantee that is analogue to that offered by leximax solutions.* Developing a meaningful notion of approximation is exactly the challenge that this paper addresses.

We provide a relaxation of the semantic definition that we term *ϵ-tradeoff leximax* (Definition 7) that is always guaranteed to exist, and while it is not equivalent to the notion presented in [10], in Theorem 11, we show that it is equivalent to a stronger variant of their definition that we call *ϵ-recursive leximax* (Definition 10). The algorithms of [10] have the potential to slightly mis-estimate the maxmin values for different groups, and therefore are only guaranteed to output approximate leximax solutions. The type of mis-estimations that may arise are actually more constrained than the full class of errors their weaker notion of approximation allows for. In particular, solutions outputted by their algorithms actually satisfy our stronger notion of ϵ-recursive leximax.

Past explorations of lexicographic maximality have mostly concentrated on finding *exact* leximax solutions. In the design of algorithms, approximations are usually viewed as alternative solutions that are "almost as good" as the exact solution and that are computed in settings where it is difficult to efficiently find exact solutions. In this paper we suggest that in some cases, we may prefer to consider an approximate notion of lexicographic maximality rather than its exact counterpart. In particular, exact leximax solutions may be highly dependent on small variations in the utility of less-well-off groups. For example, a solution where all groups receive 0.01 utility would be preferred by the exact leximax objective over a solution where one group receives 0 utility and all others receive a utility of 1, even though this second solution gets much higher utility for the majority of groups while only decreasing the utility of a single group by a tiny amount. We explore well-defined ways where approximation can benefit stakeholders and suggest a notion of approximation that is stronger than the ϵ-recursive leximax notion mentioned above that we term *ϵ-significant recursive leximax approximation* (Definition 12) that identifies solutions that ignore tiny variations in utility and identifies only solutions that are leximax due to significant increases in utility. In Theorem 14, we give a more formal characterization of the benefits drawn from considering ϵ-significant recursive leximax solutions rather than just any ϵ-recursive leximax solution.

A third motivator for our study of leximax approximations is how robust leximax solutions may be to small amounts of noise in the estimates of group utility. We show that when calculated in a noisy setting, our relaxed semantic notion (ϵ-tradeoff leximax) is not guaranteed to still be ϵ-tradeoff leximax, however it is guaranteed to satisfy the weaker notion of approximation defined in [10]. On the other hand, in Lemma 18 we show that we can define a stronger variant of the semantic notion that guarantees a solution will be ϵ-tradeoff leximax in the noisy setting, but it has the disadvantage that such solutions may not always exist. We also examine noise in the context of ϵ-significant recursive leximax solutions, and show that when such solutions are calculated in the presence of noise, they are somewhat robust to noise as they imply a slightly weakened variant of ϵ-significance (Lemma 19).

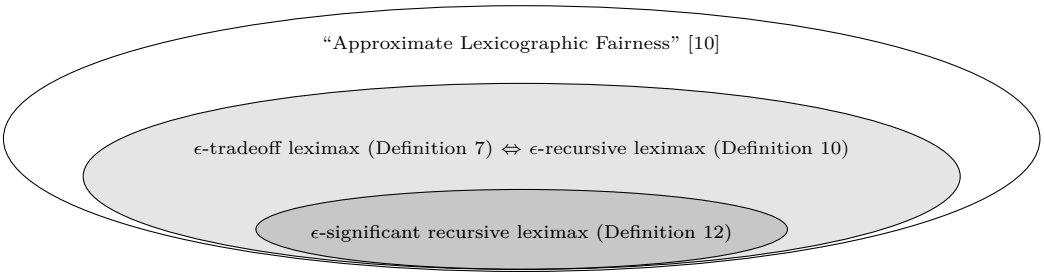

Figure 1 Relations between the different notions of leximax approximation discussed here and in [10].

Figure 1 summarizes the various notions of approximate lexicographical maximality and how they relate to one another. All of our approximate notions are defined with respect to an arbitrary class of solutions from which we'd like to pick a leximax solution. This allows our new definitions to be applied in the deterministic setting, where each solution would represent a particular cohort, or a randomized setting, where each solution corresponds to a distribution over cohorts and utilities are given in expectation.

1.2 Algorithms for Leximax Cohort Selection

In data selection, recruiting, and civic participation settings where a representative cohort is desired, the goal of representation is juxtaposed with the constraint of selecting a small representative set. There can be tension between selecting a cohort small enough for the resources available but large enough to represent as much of the population as possible. A lexicographically maximal solution is particularly salient in a representative cohort problem because it guarantees inclusion for the worst-off-groups while optimizing for the utility of all groups. We consider a model where how well each group or individual in the population is represented by a cohort candidate is given by a utility function. While there are many different ways a cohort or committee in power might make decisions or influence outcomes, we consider a linear setting where the utility a group derives from a cohort is the sum of utilities derived from each member of the cohort. Approximate notions of lexicographical maximality are of particular interest in this setting since estimating utilities that describe representativeness is difficult and might be noisy in practice.

Diana et al. [10] give a convex formulation of approximate lexicographical fairness and an oracle-efficient algorithm to solve general leximax convex programs. For our cohort selection setting specifically, we leverage the linearity across decision variables to find a polynomial time algorithm (Algorithm 1) that can calculate both exact leximax solutions as well as the two approximate variants we consider, ϵ-tradeoff leximax and ϵ-significant recursive leximax (with no external oracle needed for the calculation).

The linearity of utilities across cohort members and the recursive definition of leximax gives us a sequence of linear programs where the number of variables is linear in the size of the candidate pool and the number of constraints is exponential in the number of groups. In each m-th linear program, we maximize the sum of utilities of all sized-m groups which gives us an exponential number of constraints; rendering the linear program too big to solve via generic LP solvers. We circumvent this difficulty by creating a separation oracle (Algorithm 2) which tests the sum of utilities of the m worst off groups efficiently, giving us a polynomial time algorithm overall (Lemma 16). We can use the same approach to efficiently find ϵ-tradeoff leximax or ϵ-significant recursive leximax solutions by modifying the lower bound constraints on the sum of group utilities.

The output of our algorithm allows a randomized approach for selecting a cohort of expected size k that guarantees leximax utilities in expectation. We can also round our algorithm output to a solution of size exactly k where the expected utility across groups is leximax[1]. We focus on this distributional setting for our algorithms for a few key reasons:

- **Tractability.** If we wanted to instead find a deterministic cohort of exactly size k by finding a lexicographically maximal integer solution, the problem becomes hard. By showing that the problem of finding the exact integer lexicographically maximal cohort solves the NP-hard problem of Minimum Hitting Set, we show that finding a solution as well as approximating the number of groups with non-minimum utility within a factor of $(1 - 1/e) + o(1)$ is NP-Hard (Lemma 20).
- **Fair Arbitration between Solutions.** It is very possible that two lexicographically maximal deterministic solutions may provide wildly different utility values for a particular group. As an example, consider choosing between a cohort that provides maximum utility to Group A, but zero utility to Group B, and another cohort that provides zero utility to Group A but maximum utility to Group B. Both cohorts are a lexicographic maximum, however selecting a deterministic solution requires us to decide whether the solution should favor Group A or B. A distributional approach gets rid of this difficult decision because the randomized approach itself guarantees that we are providing both A and B a fair chance at high utility.

There are many different potential approaches to randomly selecting a cohort in the distributional setting. We choose to use a randomized approach to selection that includes or excludes each potential cohort member independently with probability outputted by the algorithm. Such an approach offers the following benefits:

- **Simple Sampling Procedure.** Rather than outputting an arbitrary and potentially complicated distribution over cohorts that is difficult to sample from, the output of our algorithm is a single vector of marginal selection probabilities for each potential candidate. Our approach still results in a cohort with expected size k, but provides an easy way to sample cohorts, and as discussed in the final bullet point, gives better guarantees about the utility groups can expect to receive in practice. We also describe a rounding approach that results in cohorts of size exactly k that are still leximax in expectation.
- **Better Concentration Guarantees for Some Natural Settings.** While a distributional leximin solution may give groups better utility guarantees in expectation, it comes with the caveat that individual runs of the randomized solution may still result in cohorts where groups receive utility that is far below their expected utility. In an extreme case, a distributional solution that guarantees all groups 0.5 utility might be achieved by choosing uniformly between solutions that provide maximum and zero utility. When the size of the cohort is large enough, our approach to randomized choice guarantees that groups receive utility near their expectation with high probability because we consider each cohort member independently, rather than outputting an arbitrary joint distribution over potential cohort members (Lemma 15).

[1] For further rounding details, see discussion in Section C.

1.3 Our Contributions

To summarize, we provide the following contributions:

1. Define a new semantic notion of leximax approximation that is always guaranteed to exist and show that it is equivalent to an algorithmically-inspired notion of approximation that is stronger but related to the one defined in [10].
2. Investigate stricter notions of approximation that identify significantly leximax solutions that can be achieved by ignoring small variations in utility.
3. Explore how our new notions of approximation behave in settings where the group utilities may be reported with some small amount of additive noise.
4. Provide polynomial time algorithms for computing exact and approximate leximax distributions over cohorts with linear utility functions.
5. Show that the alternative goal of computing deterministic cohorts in our setting is NP-hard, and moreover approximating the number of groups with non-minimum utility is also NP-hard.

▶ Note 1. Due to space constraints, all proofs except the proof of Lemma 16 (Appendix A) have been omitted, but can be found in the full version of this paper.

1.4 Related Work

Fair and diverse selection has become a prominent area of interest in algorithmic and machine learning fairness communities. In the setting of selecting representative data, prior works define metrics for diversity [24], and give algorithms for diverse data selection and summarization [7, 18]. For selecting individuals from a larger pool, prior works on cohort selection and multi-winner elections have studied individual guarantees of fairness [2] as well as group parity goals of diversity [5, 8, 27]. Other works have examined how bias and variance may affect different groups differently during a selection process and fairness amounts to remedying implicit bias and variance in the selection process for different groups of individuals [12, 17]. Parity or proportional diversity approaches to cohort selection assume the correct amount of representation for each subgroup is known and thus fairness can be achieving a predefined level of diversity.

When there is no "merit" function to guide a selection process, cohort selection can also been seen as a representation problem. Diversity is the goal of a central decision maker while representation is the objective of each group in the population when selecting a cohort. Instead of modeling overall welfare based on the number of representatives from each group, our work considers the welfare of each group based on how representative each cohort member is for that group. Since how well a cohort serves each group in a population cannot be summarized by a single value, a natural direction is to examine the utilities of all groups of a given cohort that has been selected from a general population. Lexicographical fairness emerges as a reasonable notion of fairness that guarantees Pareto optimality in this setting of multiple objectives or losses. Flanigan et al. [13] give an algorithm for recruiting "citizen's assemblies" based on sampling from a distribution over representative panels that are generated from leximax selection probabilities over citizens in the population. Our work looks at selecting a representative cohort from a pool of candidates rather than the underlying population which allows a more general model where each member or group in the population has a utility vector describing its utility for each candidate that is being considered for the cohort. Furthermore, we optimize for leximax utilities for each group of interest rather than leximax sample probabilities for each individual in the population.

In telecommunication network design, min-max fairness (MMF) is an important solution concept to lexicographically maximize fractional flow for all parties [1, 25, 26]. An adjacent problem of lexigraphically maximal flows where there are multiple sinks has also been studied and a polynomial time algorithm exists for finding fractional flow [22, 23]. The problem of finding a leximax routing for an unsplittable flow along a network is NP-Complete but finding a 2-approximation is possible [16]. An approximate solution here means that it is not possible to improve a group without decreasing the utility of another group that is more than a factor of 2 worse.

Lexicographically maximal solutions have also been studied in other domains including bottleneck combinatorial optimization problems [6, 9], sampling actions for repeated games [3], allocation of classrooms [19] as well as indivisible goods more generally [14]. It is important to note that unlike the leximax allocation problem, there is no limit on the number of groups gaining utility from the same candidate being included in a cohort or allocated set. Most recently, leximax empirical risk minimization for classification has also been studied [15, 20, 21].

2 The Leximax Objective

In this paper, we focus on approaches to selecting *lexicographically maximal* (or leximax) representative cohort solutions. We consider a setting in which we'd like to select a solution S from a set of potential solutions $\mathcal{S}$ such that S is a good representation of some set of key (potentially overlapping) subgroups $\mathcal{G} = \{G_1, ..., G_m\}$. We measure degree of representation via a utility function $u : \mathcal{S} \times \mathcal{G} \to [0, 1]$. Ideally, we'd like to select a cohort such that every subgroup is guaranteed to have high utility. However, this may be impossible to achieve in certain settings, such as when the utility functions of two groups are in opposition. Unlike maximizing total welfare, which may result in solutions that neglect the welfare of certain groups or seeking to equalize utilities across groups, which may artificially cap the utility some groups can achieve, lexicographically maximal solutions extend the goal of the classic maxmin objective by seeking to maximize the utility of the worst-off group, and then seeking to maximize the utility of the second-worst-off group subject to this worst-off group's value, etc. This results in a solution concept that seeks to give the best guarantee possible for every key group, rather than just the worst-off.

We now formally define the leximax objective.

▶ **Definition 2.** *Given two vectors u and v in $\mathbb{R}^m$, we say that u is lexicographically greater than v, or $v \preceq u$, if and only if there exists some i such that for all $j \leq i$ we have $v_j = u_j$, and either $i = m$ or $u_{i+1} > v_{i+1}$.*

Applying this definition to the set of sorted group utility vectors obtained from every possible solution gives us a total ordering on these vectors. A leximax solution is any vector that is maximal according to this ordering. In many portions of this paper, in order to reason about the contents of these sorted vectors, we will care about the utility that the ith worst-off group receives from a particular solution S. We denote this with the bracketed notation $u(S, G_{[i]})$.

▶ **Definition 3.** *Given a set of potential solutions $\mathcal{S}$ and groups $\mathcal{G}$, we say that a solution $S \in \mathcal{S}$ is lexicographically maximal (leximax) if for any other solution S', we have $\langle u(S', G_{[i]})\rangle_{i=1}^{m} \preceq \langle u(S, G_{[i]})\rangle_{i=1}^{m}$.*

Intuitively, when we seek to find a lexicographically maximal solution, we try to do the best we can for the worst-off group, and then within these potential solutions try to do the best we can for the second-worst-off group, etc. Note that under this definition, groups may

achieve varying utilities for different lexicographically maximal solutions, however the vector of sorted group utilities will be unique for any leximax solution. When the solution class is convex and compact and the utility function is continuous with respect to this class, a particular group receives the same utility under any leximax solution.

An attractive feature of lexicographically maximal solutions is that they have an equivalent definition that gives a semantic understanding of the solutions identified by the goal in Definition 3. We call this notion *tradeoff leximax*.

▶ **Proposition 4.** *Given a set of solutions $\mathcal{S}$ and groups $\mathcal{G}$, $S \in \mathcal{S}$ is lexicographically maximal if and only if for any S' and $i \in [m]$ such that $u(S', G_{[i]}) > u(S, G_{[i]})$, there exists some $j < i$ such that $u(S, G_{[j]}) > u(S', G_{[j]})$.*

This equivalent definition of lexicographic maximality offers an appealing re-interpretation of this objective: a solution is optimal if increasing the utility of any particular group would result in decreasing the utility of a worse-off group.

3 Approximations of Leximax-Optimal Solutions

While the leximax objective's goal of doing the best we can for every group is attractive, one potential downside is that the set of leximax-optimal solutions can be incredibly sensitive to small variations in the utility received by certain groups. We consider the following example that illustrates this phenomenon:

▶ **Example 5** (Sensitivity of leximax-optimal solutions)**.** Consider a simple setting as in Figure 2 in which we have two groups, $\mathcal{G} = \{G_1, G_2\}$, and would like to decide between two potential solutions $\mathcal{S} = \{S_1, S_2\}$. The utilities for each group and each solution are defined as $u(S_i, G_j) = U_{ij}$ where $U \in [0, 1]^{\mathcal{S} \times \mathcal{G}}$ is defined as follows:

$$U = \begin{bmatrix} 0 & 1 \\ 0.01 & 0.01 \end{bmatrix}.$$

Clearly the only leximax solution is S_2 (with sorted utility vector $(0.01, 0.01)$), because the worst-off group has value 0.01 rather than receiving 0 utility as it does in S_1 (which has a sorted utility vector of $(0, 1)$.

However, if we allow for the possibility that the utility estimates are off by even a tiny amount such as 0.01, suddenly S_1 is also a plausibly leximax solution despite having a completely different value for the second-worst-off group.

Example 5 is notable in that it demonstrates how small variations in the utilities of groups can lead to drastic changes with respect to the types of leximax solutions that are considered optimal. In settings where utilities may be reported with some estimation error, it is therefore incredibly important to consider how these errors might affect how the output optimal solution compares to the true leximax solution that would have been produced given completely accurate utilities.

Moreover, even when the utilities are believed to be accurate, it may be useful to consider solutions that are not exactly leximax, but are leximax when small variations in the utility are ignored. Example 5 is a situation where the exact leximax offers a tiny improvement in the worst-off group at the cost of a huge decrease in the utility of the second-worst-off group. A practitioner who views utility differences of less than 0.05 as insignificant might prefer S_1 as the only *significantly* lexicographically maximal solution because the worst-off groups between S_2 and S_1 receive comparable utility while the second-worst-off group is significantly better off under S_1.

The search for plausibly exact lexicographic solutions given the potential for some amount of estimation error as well as the need for significantly maximal lexicographic solutions even when working with exact utility values motivates our study of new approximate leximax notions. In this section, we introduce two such notions: first, we introduce a semantic notion of approximate leximax that relaxes the standard leximax definition to consider additional solutions that may be plausibly leximax. The second notion we introduce here seeks solutions that are leximax if only "significant" improvements are considered (as in the discussion above). Unlike the first notion, the notion of significantly leximax solutions is not a strict relaxation of leximax and may not include the exact leximax solution in some cases.

3.1 Relaxations of the Leximax Objective

3.1.1 Elementwise Approximation

The most naive approach to approximation would be to require that the element-wise distance between the sorted utility vectors of the true lexicographically maximal solution and the approximate solution be small:

▶ **Definition 6** (Element-wise leximax approximation). *Given a set of m groups $\mathcal{G}$ and a set of potential solutions $\mathcal{S}$, let ℓ be the sorted vector of utilities attained by any leximax solution. We say that a solution $S \in \mathcal{S}$ is an α-element-wise leximax approximation iff* $\max_{i\in[m]}\{\ell_i - u(S, G_{[i]})\} \leq \alpha$.

While attractive in its simplicity, [10] observe that in certain contexts, such a definition may be stricter than we can hope for. In particular, if the leximax solution is being computed recursively, small estimation errors in the values of the worst-off group's utility can greatly effect the difference between the utility of better-off groups in a lexicographically maximal solution compared to a solution that maximizes group utilities based off of this incorrect value. Thus, we turn our attention to weaker notions of approximation.

3.1.2 Tradeoff Approximation

We introduce a new notion of approximation that is a natural relaxation of the semantic interpretation of leximax solutions provided by the *tradeoff leximax* objective discussed in Proposition 4.

▶ **Definition 7** (ϵ-tradeoff leximax). *Given a set of m groups, $\mathcal{G}$, and a set of potential solutions, $\mathcal{S}$, a solution $S \in \mathcal{S}$ is ϵ-tradeoff leximax if for any S' and i such that $u(S, G_{[i]}) < u(S', G_{[i]}) - \epsilon$, there exists a $j < i$ such that $u(S, G_{[j]}) > u(S', G_{[j]})$.*

Intuitively, this definition guarantees that if we can find some other solution that does a lot better on some particular group, then this new solution must also decrease the utility of some worse-off group.

ϵ-tradeoff leximax provides an appealingly simple relaxation of the semantic interpretation of exact leximax solutions. However, slight variations of this definition, also natural relaxations of leximax, will result in definitions where solutions are not guaranteed to exist. We explore this in the following example:

▶ **Example 8** (Altered versions of ϵ-tradeoff leximax may not have any solutions.). We define a class of alternative tradeoff definitions that we term (ϵ_1, ϵ_2)-significant tradeoff leximax for reasons that will become clear in Section 3.2 as follows:

▶ **Definition 9** ((ϵ_1, ϵ_2)-significant tradeoff leximax). *Given a set of m groups, $\mathcal{G}$, and a set of potential solutions, $\mathcal{S}$, a solution $S \in \mathcal{S}$ is (ϵ_1, ϵ_2)-significant tradeoff leximax for any $\epsilon_1, \epsilon_2 \geq 0$ if for any S' and i such that $u(S, G_{[i]}) < u(S', G_{[i]}) - \epsilon_1$, there exists a $j < i$ such that $u(S, G_{[j]}) > u(S', G_{[j]}) + \epsilon_2$.*

When $\epsilon_1 = \epsilon$ and $\epsilon_2 = 0$, this notion is equivalent to ϵ-tradeoff leximax. When $\epsilon_2 > 0$, the definition requires that any increase by more than ϵ_1 result in a decrease of more than ϵ_2 in a worse-off group.

However, we demonstrate that for $\epsilon_1, \epsilon_2 > 0$, no solution may exist. Consider $\epsilon_1 = \epsilon_2 = \epsilon$ for the setting depicted in Figure 3 where we have two groups and four potential solutions with utilities defined as $u(S_i, G_j) = U_{ij}$, for

$$U = \begin{bmatrix} 0 & 0.5 + 6\epsilon \\ \epsilon/2 & 0.5 + 4\epsilon \\ \epsilon & 0.5 + 2\epsilon \\ 3\epsilon/2 & 0.5 \end{bmatrix}.$$

Where we assume ϵ is sufficiently smaller than 0.5. Under these utilities, S_4 cannot be (ϵ, ϵ)-significant tradeoff leximax because S_3 improves by more than ϵ in G_2 while only decreasing G_1 by $\epsilon/2$. Similarly, S_3 and S_2 cannot be (ϵ, ϵ)-significant tradeoff leximax due to the existence of S_2 and S_1, respectively. This means that S_2, S_3, S_4 all cannot be (ϵ, ϵ)-significant tradeoff leximax. However, we see that S_4 improves by more than ϵ over S_1 in G_1, so S_1 also cannot be (ϵ, ϵ)-significant tradeoff leximax. We conclude that no potential solution satisfies this definition[2].

Computing tradeoff approximations recursively

The definition of ϵ-tradeoff leximax is only useful if we can compute ϵ-tradeoff leximax solutions efficiently. To show that this is possible, we relate ϵ-tradeoff leximax to a different notion of leximax approximation that arises from a natural algorithmic approach and is closely related to the notion of leximax approximations introduced in [10].

Consider the following approach to computing an exact leximax solution, which follows its definition: Compute the maximum value that can be guaranteed to the worst-off group, then calculate the maximum value that can be guaranteed to the second worst-off group subject to this value, and then recurse on the third, fourth, fifth, etc. until the values for all m groups are fixed and a solution is found.

However, what if our algorithm is not completely accurate at each step? Introducing some amount of estimation error at each step of the recursion may result in selecting a solution that isn't exact leximax, but can considered approximately leximax because it arose from small estimation errors in our algorithm. We call such solutions ϵ-recursive leximax, and define them as follows:

▶ **Definition 10** (ϵ-recursive leximax). *Given a set of m groups, $\mathcal{G}$, a set of potential solutions $\mathcal{S}$, and a choice of allowable "slack" $\vec{\alpha} = (\alpha_1, ..., \alpha_m)$ with $\alpha_i \in \mathbb{R}_{\geq 0}$, recursively define the sets of solutions $\mathcal{S}_0^\alpha, ..., \mathcal{S}_m^\alpha \subseteq \mathcal{S}$ such that $\mathcal{S}_0^\alpha := \mathcal{S}$ and for each $i = 1, ..., m$,*

$$\mathcal{S}_i^\alpha = \{S \in \mathcal{S}_{i-1}^\alpha : u(S, G_{[i]}) \geq \max_{S' \in \mathcal{S}_{i-1}^\alpha} u(S', G_{[i]}) - \alpha_i\}.$$

We say that $S \in \mathcal{S}$ is an ϵ-recursively approximate leximax solution if there exists an $\vec{\alpha}$ with $\max_{i \in [m]} \alpha_i \leq \epsilon$ such that $S \in \mathcal{S}_m^\alpha$.

[2] This example was not tied to the specific choice of $\epsilon_1 = \epsilon_2 = \epsilon$. Similar examples exist for other choices.

Our definition of ϵ-recursive leximax is a stronger variant of the definition of approximation used in [10]. Most importantly, the definition presented in [10] is less strict because it allows for the choice of allowable slack to depend on each solution. However, the solutions outputted by their algorithms actually achieve the stronger notion presented here. Unlike the weaker version, which is only implied by ϵ-tradeoff leximax, we can show that ϵ-recursive leximax and ϵ-tradeoff leximax are equivalent.

In this definition, the choice of slack, $\vec{\alpha} \in [0, \epsilon]^m$, determines the amount of estimation error at each step. We use this $\vec{\alpha}$ to recursively construct the sets $\mathcal{S}_i^\alpha$ in the same way they would be calculated had we applied a recursive approach to calculating a leximax solution but under-estimated the maximum value by α_i at the ith step for each $i = 1, ..., m$.

Unlike our ϵ-tradeoff leximax notion of approximation, ϵ-recursive leximax provides a natural algorithmic interpretation of approximate solutions which allows efficient approaches to computing ϵ-recursive leximax solutions with respect to a particular choice of slack, as we do in Section 4[3]. Fortunately, we can actually show that these two notions of approximation are equivalent, which means that we can also efficiently compute ϵ-tradeoff leximax solutions.

▶ **Theorem 11.** *For any set of groups, $\mathcal{G}$, and solutions, $\mathcal{S}$, the set of ϵ-tradeoff leximax solutions is equivalent to the set of ϵ-recursive leximax solutions.*

3.2 Significantly Leximax Solutions

ϵ-tradeoff leximax solutions are strict relaxations of the exact leximax objective. Any leximax-optimal solution will also be ϵ-tradeoff leximax and will also be ϵ-recursive leximax for any $\epsilon \geq 0$ (by simply selecting the allowable slack to be $\alpha_i = 0$ for all $i \in [m]$). Similarly, any ϵ-tradeoff (resp. recursively) approximate solution will also be ϵ'-tradeoff (recursively) approximate for any $\epsilon' \geq \epsilon$.

In this section, we introduce a modified notion of ϵ-recursive leximax that is not a relaxation of the exact leximax objective but rather tries to get significant improvements in the quality of solutions, using the allowed slack. This notion constrains the choices of slack so that solutions considered leximax due to only insignificant improvements in the utility of worse-off groups are ignored. Here, the only slack considered is where all allowable slack values are set to exactly ϵ, rather than some value that is at most ϵ.

▶ **Definition 12** (ϵ-significant recursive leximax)**.** *Given a set of groups $\mathcal{G}$ with $|\mathcal{G}| = m$ and a set of potential solutions $\mathcal{S}$, recursively define the sets of solutions $\mathcal{S}_0^\epsilon, ..., \mathcal{S}_m^\epsilon \subseteq \mathcal{S}$ such that $\mathcal{S}_0^\epsilon := \mathcal{S}$ and for each $i = 1, ..., m$,*

$$\mathcal{S}_i^\epsilon = \{S \in \mathcal{S}_{i-1}^\epsilon : u(S, G_{[i]}) \geq \max_{S' \in \mathcal{S}_{i-1}^\epsilon} u(S', G_{[i]}) - \epsilon\}.$$

We say that $S \in \mathcal{S}$ is ϵ-significant recursive leximax if $S \in \mathcal{S}_m^\epsilon$.

Why does this make sense as a way to identify significant solutions? Intuitively, setting every slack value to the maximum possible ϵ requires that the valid solutions be leximax with respect to the larger set of potential solutions when some error term is allowed, rather than putting a lot of weight on small differences in earlier groups. We present the following example to see this in practice:

[3] [10] give algorithms that calculate ϵ-recursive leximax solutions because their approach estimates each sequential maxmin value to within ϵ of its true value, though the notion of efficiency that they achieve does not exactly correspond to polynomial-time algorithms. We provide an alternative polynomial-time algorithm for the cohort selection setting that leverages linear group utilities to offer a more efficient approach.

▶ **Example 13** (Significantly recursive approximations). Consider two groups and two solutions as in Figure 4 with utilities

$$u(S_1, G_1) = \epsilon, u(S_2, G_1) = 0, u(S_1, G_2) = 0.5, u(S_2, G_2) = 1.$$

Both S_1 and S_2 are ϵ-recursive leximax approximations. If we set $\alpha_1 < \epsilon$, then S_1 because the only acceptable solution and thus an ϵ-recursive leximax-approximate solution. If we set $\alpha_1 = \epsilon$, S_2 becomes an ϵ-recursive leximax-approximate solution. We would expect a satisfying significant approximation notion to identify S_2 as the only ϵ-significant approximation because it's not too far below S_1 on the worst-off group, but does much better on the second-worst-off group. An ϵ-significant recursive leximax approximation does give us this separation between S_1 and S_2, because while both S_1 and S_2 are included in the first-level of recursion, $\mathcal{S}_1^\epsilon$, S_1 is too far below the maximum to be included in $\mathcal{S}_2^\epsilon$, so S_2 is the only ϵ-significant recursive leximax approximation in this example.

So far, we have been rather loose in arguing about why the solutions identified as ϵ-significant recursive leximax might be preferred over exact leximax or the more general class of ϵ-recursive leximax solutions. We offer a more formal characterization here, but begin by taking a step back to reframe what the contents of the recursively defined sets from Definition 10, $\mathcal{S}_1^\alpha, ..., \mathcal{S}_m^\alpha$ for some choice of slack $\vec{\alpha}$, can tell us about potential leximax solutions.

Intuitively, $\mathcal{S}_i^\alpha$ contains all solutions that, with respect to the first i groups, could feasibly be solutions that are ϵ-recursive leximax allowing for a slack of $\vec{\alpha}$, and are guaranteed to be within ϵ of the first i coordinates of any final ϵ-recursive leximax solution with respect to $\vec{\alpha}$, i.e. any $S \in \mathcal{S}_m^\alpha$.

This means that looking at the maximum utility achieved by any solution in each recursive group, $\langle \max_{S \in \mathcal{S}_i^\alpha} u(S, G_{[i]}) \rangle_{i=1}^m$ gives us a sense of the type of solution that results from allowing $\vec{\alpha}$ as slack. While there may not exist a $S' \in \mathcal{S}_m^\alpha$ such that $\langle u(S', G_{[i]}) \rangle_{i=1}^m = \langle \max_{S \in \mathcal{S}_i^\alpha} u(S, G_{[i]}) \rangle_{i=1}^m$, we are guaranteed that any $S' \in \mathcal{S}_m^\alpha$ will be elementwise within ϵ of this vector of maximums.

We can show that out of all possible choices of slack, the one used by the definition of ϵ-significant recursive leximax, $\vec{\alpha} = (\epsilon, ..., \epsilon)$ results in the best-possible sequence of maximum set values (i.e. it will be lexicographically greater than the maximums attained via any other choice of slack). In other words, this backs up the motivation behind our definition of ϵ-significant recursive leximax in that it promises us that any ϵ-significant recursive leximax solution will be elementwise within ϵ of the lexicographically best solution we could possibly hope for under an optimal choice of slack.

▶ **Theorem 14** (Leximax properties of ϵ-significant recursive leximax). *Given a set of groups, $\mathcal{G}$, and solutions, $\mathcal{S}$, let $\mathcal{S}_1^\epsilon, ..., \mathcal{S}_m^\epsilon$ be the recursively defined sets constructed with a slack of ϵ at each step, as used in the definition of ϵ-significant recursive leximax, and for any $\vec{\alpha} \in \mathbb{R}_{\geq 0}^m$, let $\mathcal{S}_1^\alpha, ..., \mathcal{S}_m^\alpha$ be the sets that arise when the amount of allowable slack at each level is set according to $\vec{\alpha}$. Then, for any $\vec{\alpha} \in \mathbb{R}_{\geq 0}^m$, we have*

$$\langle \max_{S \in \mathcal{S}_i^\epsilon} u(S, G_{[i]}) \rangle_{i=1}^m \succeq \langle \max_{S \in \mathcal{S}_i^\alpha} u(S, G_{[i]}) \rangle_{i=1}^m.$$

In other words, the vector of maximums attained in each $\mathcal{S}_i^\epsilon$ is lexicographically maximal compared to any other choice of slack of size at most ϵ.

Theorem 14 tells us that out of all the ways we could identify approximate leximax solutions that ignore variations of less than ϵ, an ϵ-significant solution is guaranteed to be element-wise within ϵ on the lexicographically maximal best-possible guarantee we can give for each group at each level of recursion.

Ideally, we could obtain a similar notion to ϵ-significant recursive leximax with a satisfying semantic meaning as for ϵ-recursive leximax by modifying our definition of ϵ-tradeoff leximax so that any solution that improves the ith group by more than ϵ must also decrease some worse-off group by more than ϵ. However, as we saw in Example 8, modifying the original tradeoff definition in this way surprisingly results in an overly strict notion due to some instability arising from the pairwise comparisons that tradeoff approximations rely on. In particular, solutions that satisfy this notion may not exist. Note that in Example 8, no solution satisfied (ϵ, ϵ)-significant tradeoff leximax, which is equivalent to the modified definition suggested here, but S_2 is an ϵ-significant recursive leximax approximation and S_2, S_3, S_4 are all valid ϵ-recursive leximax solutions.

4 Solutions via Linear Programming

As discussed in Section 1.2, we provide efficient algorithms for a particular natural choice of cohort selection setting. In particular, we consider modeling utility as the sum of the utilities that a subgroup draws from each individual member of the selected cohort, and rather than outputting a lexicographically maximal cohort, we output a lexicographically maximal vector of marginal selection probabilities that provides leximax utility in expectation.

4.1 Problem Setting

We begin by discussing our choice of utility function and randomized selection approach in more detail.

4.1.1 Linear Utility Function

Let $\mathcal{C}$ be a set of potential committee members of size n. We assume that each subgroup $G_j \in \mathcal{G}$ has a value for each individual committee member $c_i \in \mathcal{C}$, denoted by $v_{ij} \in [0, 1]$.

When we choose our set of solutions to be $\mathcal{C}^{(k)}$, the set of all cohorts of size k, these values can now be combined to give a group's utility for any particular cohort as the sum of its values for the cohort members. Given a cohort $C = \{c_1, ..., c_k\} \in \mathcal{C}^{(k)}$ and subgroup $G_j \in \mathcal{G}$, this utility function can be written formally as

$$u(C, G_j) = \sum_{i=1}^{k} v_{ij}$$

This linear utility can easily be extended to the randomized case. Assuming $\mathcal{C}$ has size n, any vector of individual assignment probabilities $D = \{x_1, \ldots, x_n\} \in \mathcal{D} := [0, 1]^n$, called *marginal (selection) probabilities*, provides an approach to randomly selecting a cohort of candidates from $\mathcal{C}$ where each c_i is included in the cohort with probability x_i, independent of the other candidates. The expected utility of a particular group G_j over a distribution $D \in [0, 1]^n$ is then

$$u(D, G_j) = \sum_{i=1}^{n} x_i v_{ij}.$$

We will restrict our search to marginal distributions that output a cohort with expected size k ($\sum_{i=1}^{n} x_i = k$).

4.1.2 Randomized Selection Approach

Our algorithms output a vector of marginal selection probabilities $D = \{x_1, ..., x_n\} \in \mathcal{D} := [0, 1]^n$, such that when each cohort member c_i is independently included in the cohort with probability x_i, we get a cohort of size k in expectation such that the vector of expected utilities is lexicographically maximal. This independent sampling procedure provides a simple way to randomly select a cohort.

This distributional approach to selection renders the problem tractable, while we show in Appendix D that finding deterministic leximax solutions is NP-hard. Moreover, it provides a fair way to get around the issue that in a deterministic setting, there may be multiple leximax cohorts that each favor a different subgroup.

It's worth noting that this approach to cohort selection only gives a cohort with *expected* size k. While such a selection procedure may be fine in situations where the desired size of the final cohort is somewhat flexible, sometimes it may be critical to get a cohort of size exactly k. In Appendix C, we discuss a dependent rounding scheme that can be used to sample a cohort of size exactly k with utilities that are still leximax in expectation.

In general, cohorts sampled from arbitrary leximax distributions are not guaranteed to provide groups with utility near their expected value. However, our choice of selection procedure guarantees that groups receive near-expected utility with high probability when the size of the selected cohort is large enough.

▶ **Lemma 15.** *Consider an arbitrary group G_j and a lexicographically maximal vector of marginal selection probabilities $D \in \mathcal{D}$ (with respect to the linear utility function defined above and with expected size k).*

Then, for any $\delta > 0$, we have

$$\Pr_{C \sim D}[U(C, G_j) < U(D, G_j) - \delta] < e^{-2\delta^2/n}.$$

(Where $n := |\mathcal{C}|$ is the number of potential cohort members.)

To contextualize this result, if we consider some group G_j that is expected to get about half of their maximum possible utility for a leximax solution when $k = 50$ and $n = 100$ (so G_j is expected to get 25 utility), then they are guaranteed to get at least half their expected utility more than 95% of the time. In comparison, an arbitrary leximax distribution can potentially only guarantee that G_j gets more than half their expected utility with probability 1/3. These concentration guarantees also hold for cohorts of size exactly k outputted by our suggested rounding approach. More details can be found in Section C.

Having explained and justified our choice of utility function as well as randomized selection approach, we now present our algorithms that calculate exact and approximate leximax solutions in this setting.

4.2 Leximax distribution over committee members

To find a marginal distribution over each potential committee member in $\mathcal{C}$, we break up the problem into multiple, recursively-defined subproblems to uncover the ranking of subgroup utilities in the leximax optimal solution as well as their optimal values.

Balan et. al [3] approach this problem by reducing the domain of solutions in each level of optimization. They choose the $(i + 1)$-th subgroup to be the subgroup that least-constrains the domain of potential leximax solutions. Overall, their approach finds a leximax-optimal marginal distribution over potential committee members that requires $O(|\mathcal{G}|)$ calls to a linear program at each of the $|\mathcal{G}|$ iterations, giving us $O(|\mathcal{G}|^2)$ total calls. However, this approach

of limiting the domain of the possible solutions requires fixing an order of worst off groups in every iteration. When approximate notions of leximax are introduced, there can be multiple possible orderings of groups to consider.

We suggest finding the leximax distribution over individuals as a series of linear programs with a linear number of variables and a number of constraints that increases from linear to exponential as the series progresses. In the first LP, we are finding the maxmin utility γ_1 using the values v_{ij} that each group has for individual cohort candidates:

$$\begin{array}{ll} \text{maximize}_{x,\gamma_1} & \gamma_1 \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & 0 \le x_i \le 1 \\ & \sum_{i=1}^{n} x_i v_{ij} \ge \gamma_1 \quad j = 1, \dots, m. \end{array}$$

Once the optimal lower bound for the worst off group, γ_1^*, is found, we can then maximize the utility of the second-worse-off-group. Ogryczak et al. [26] observed that maximizing the $\gamma = (\gamma_1, \dots, \gamma_m)$ vector is equivalent to maximizing for the cumulative sum of γ_i's from $i = 1, \dots, m$. Thus, to find the leximax distribution of individuals, we optimize a series of m linear programs using the cumulative leximax values as a constraint. The m-th last LP will be as follows:

$$\begin{array}{ll} \text{maximize}_{x,\gamma_m} & \gamma_m \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & 0 \le x_i \le 1 \\ & \sum_{i=1}^{n} \sum_{G_j \in S} v_{ij} x_i \ge \sum_{s=1}^{l} \gamma_s^* \quad \forall l = 1, \dots, m, \forall S \subseteq \mathcal{G} \ s.t. \ |S| = l. \end{array}$$

Since we must ensure that the sum of utilities is above the minimum utility for all subgroups, the last constraint requires that the sum of utilities over all sized-l subsets of groups be greater than the sum of the l optimal γ^*-s (i.e. $\sum_{i=1}^{l} \gamma_i^*$) from previous iterations. This creates $\binom{m}{l}$ constraints for the l-th LP. Algorithm 1 describes the iterative process of finding a leximax distribution where in each successive problem we add additional constraints on the minimum value of the sum of utilities. In our setting of linear utilities, we can solve each linear program in polynomial time with the ellipsoid method using a polynomial-time separation oracle.

Algorithm 1 LEXIMAXCANDIDATES. Finding the leximax distribution over candidates.

Input: $v \in \mathbb{R}_{\ge 0}^{n \times m}$ values of each group for each candidate.
Output: $\{x_1, \dots, x_n\}$ leximax distribution over candidates.
Constraints $= \{\sum_{i=1}^{n} x_i v_{ij} \ge \gamma_1 \ j = 1, \dots, m; \ 0 \le x_i \le 1; \ \sum_{i=1}^{n} x_i = k\}$;
$\gamma_1^* \leftarrow \max_{x,\gamma_1} \gamma_1$ s.t. Constraints ;
for $l \in 2, \dots, m$ **do**
  Constraints = Constraints $\cup \{\sum_{i=1}^{n} \sum_{G_j \in S} v_{ij} x_i \ge \sum_{s=1}^{l} \gamma_s^* \ \forall S \subseteq \mathcal{G} \ s.t. \ |S| = l \}$;
  $\gamma_i^* \leftarrow \max_{x,\gamma_i} \gamma_i$ s.t. Constraints given $\gamma_1^*, \dots, \gamma_{i-1}^*$(previously computed);

▶ **Lemma 16.** *For n candidates and m groups, the running time of Algorithm 1 is polynomial in n and m.*

Approximate Leximax distribution over candidates

When finding approximate leximax distributions over candidates, the approach of Balan et al. [3] can no longer be applied since choosing the subgroup that least constrains the domain of potential solutions may yield multiple subgroups when the leximax objective is approximate. Thus, there is no single ordering of worst-off-groups to rely on when considering group utility. However, we can easily modify our recursive linear program (Algorithm 1) to find an an ϵ-recursive leximax solution (Definition 10) for a given "slack" vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m)$. While the first LP is the same as the exact case, we can loosen the constraints in the m-th LP as follows:

$$\begin{array}{ll} \text{maximize}_{x,\gamma_m} & \gamma_m \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & 0 \leq x_i \leq 1 \\ & \sum_{i=1}^{n} \sum_{G_j \in S} v_{ij} x_i > \sum_{s=1}^{l} (\gamma_s^* - \alpha_s) \ \forall S \subseteq \mathcal{G} \ s.t. \ |S| = l, \ l = 1, \ldots, m-1 \\ & \sum_{i=1}^{n} \sum_{G_j \in \mathcal{G}} v_{ij} x_i > \sum_{s=1}^{m-1} (\gamma_s^* - \alpha_s) + \gamma_m. \end{array}$$

For a ϵ-significant recursive leximax approximate solution, we can set all the α_i's equal to ϵ and apply algorithm 1 with modified constraints as described above.

5 Discussion and Future Work

Motivated by the problem of selecting representative cohorts, we turned to a lexicographically maximal definition of optimal representation. We investigated existing approximations of leximax fairness and introduced new definitions which consider semantic notions of noise and tradeoffs. In settings where utilities or objectives are roughly estimated and leximax fairness is desirable, the approximate notions of leximax in this paper may be useful as alternatives to exact leximax.

While we gave a polynomial time algorithm which computes a leximax distribution over a pool of candidates that is effective for both exact and approximate notions of leximax, finding an algorithm for approximation notions of leximax that is more efficient than exact algorithms remains an open problem. Furthermore, our setting of linear utilities is a natural assumption but can be extended to sub-modular or other classes of utility functions.

In another direction, our approximation notions all reason about allowing for additive amounts of error. However, considering what notions, especially those in line with ϵ-significant recursive leximax, might arise from multiplicative error could be a useful direction to explore.

Finally, we only considered how the presence of additive noise might affect our definitions, but other models of noise specific to different domains may also be considered. Noise can appear not just based on entire cohorts or distributions but also for candidates individually. Modeling how noise from individual candidates accumulate over over cohorts and distributions of candidates will vary depending on the utility function but is a promising direction to explore.

References

1 Miriam Allalouf and Yuval Shavitt. Centralized and Distributed Algorithms for Routing and Weighted Max-Min Fair Bandwidth Allocation. *IEEE/ACM Transactions on Networking*, 16(5):1015–1024, October 2008. doi:10.1109/TNET.2007.905605.

2 Konstantina Bairaktari, Huy Le Nguyen, and Jonathan Ullman. Fair and optimal cohort selection for linear utilities. *arXiv preprint*, 2021. arXiv:2102.07684.

3 Gabriel Balan, Dana Richards, and Sean Luke. Algorithms for leximin-optimal fair policies in repeated games. Technical Report GMU-CS-TR-2008-1, George Mason University, 2008.

4 Robert G Bland, Donald Goldfarb, and Michael J Todd. The ellipsoid method: A survey. *Operations research*, 29(6):1039–1091, 1981.

5 Robert Bredereck, Piotr Faliszewski, Ayumi Igarashi, Martin Lackner, and Piotr Skowron. Multiwinner elections with diversity constraints. *arXiv preprint*, 2017. arXiv:1711.06527.

6 Rainer E Burkard and Franz Rendl. Lexicographic bottleneck problems. *Operations Research Letters*, 10(5):303–308, 1991.

7 Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse dpp-based data summarization. In *International Conference on Machine Learning*, pages 716–725. PMLR, 2018.

8 L Elisa Celis, Lingxiao Huang, and Nisheeth K Vishnoi. Multiwinner voting with fairness constraints. *arXiv preprint*, 2017. arXiv:1710.10057.

9 Federico Della Croce, Vangelis Th Paschos, and Alexis Tsoukias. An improved general procedure for lexicographic bottleneck problems. *Operations research letters*, 24(4):187–194, 1999.

10 Emily Diana, Wesley Gill, Ira Globus-Harris, Michael Kearns, Aaron Roth, and Saeed Sharifi-Malvajerdi. Lexicographically Fair Learning: Algorithms and Generalization. *arXiv:2102.08454 [cs, stat]*, February 2021. arXiv:2102.08454.

11 Benjamin Doerr. *Probabilistic Tools for the Analysis of Randomized Optimization Heuristics*, pages 1–87. Springer, January 2020. doi:10.1007/978-3-030-29414-4_1.

12 Vitalii Emelianov, Nicolas Gast, Krishna P Gummadi, and Patrick Loiseau. On fair selection in the presence of implicit variance. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 649–675, 2020.

13 Bailey Flanigan, Paul Gölz, Anupam Gupta, Brett Hennig, and Ariel D Procaccia. Fair algorithms for selecting citizens' assemblies. *Nature*, 596(7873):548–552, 2021.

14 Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible goods. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 280–286. ijcai.org, 2019. doi:10.24963/ijcai.2019/40.

15 Mohammad Mahdi Kamani, Rana Forsati, James Z Wang, and Mehrdad Mahdavi. Pareto efficient fairness in supervised learning: From extraction to tracing. *arXiv preprint*, 2021. arXiv:2104.01634.

16 Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 568–578. IEEE, 1999.

17 Jon Kleinberg and Manish Raghavan. Selection problems in the presence of implicit bias. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

18 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *International Conference on Machine Learning*, pages 3448–3457. PMLR, 2019.

19 David Kurokawa, Ariel D. Procaccia, and Nisarg Shah. Leximin Allocations in the Real World. *ACM Transactions on Economics and Computation*, 6(3-4):11:1–11:24, October 2018. doi:10.1145/3274641.

20 Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR, 2020.

21 Natalia L Martinez, Martin A Bertran, Afroditi Papadaki, Miguel Rodrigues, and Guillermo Sapiro. Blind pareto fairness and subgroup robustness. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7492–7501. PMLR, 18–24 July 2021. URL: https://proceedings.mlr.press/v139/martinez21a.html.

22 Nimrod Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming: Series A and B*, 7(1):97–107, December 1974. doi:10.1007/BF01585506.

23 Nimrod Megiddo. A good algorithm for lexicographically optimal flows in multi-terminal networks. *Bulletin of the American Mathematical Society*, 83(3):407–409, 1977.

24 Margaret Mitchell, Dylan Baker, Nyalleng Moorosi, Emily Denton, Ben Hutchinson, Alex Hanna, Timnit Gebru, and Jamie Morgenstern. Diversity and inclusion metrics in subset selection. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 117–123, 2020.

25 Dritan Nace and Michal Pióro. Max-min fairness and its applications to routing and load-balancing in communication networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 10(4):5–17, 2008.

26 Wlodzimierz Ogryczak, Michal Pióro, and Artur Tomaszewski. Telecommunications network design and max-min optimization problem. *Journal of telecommunications and information technology*, pages 43–56, 2005.

27 Candice Schumann, Samsara N Counts, Jeffrey S Foster, and John P Dickerson. The diverse cohort selection problem. *arXiv preprint*, 2017. arXiv:1709.03441.

28 Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 588–597. IEEE, 2001.

A Proof of Lemma 16

Proof. We run m LPs in total. For each LP, the running time is the number of steps the ellipsoid algorithm takes multiplied by the time per iteration. For an efficient implementation, the ellipsoid algorithm needs (1) a feasible initial solution and (2) a polynomial-time separation oracle.

(1) For an initially feasible solution, set $\gamma_1 = 0$ and all $x_i = k/n$ for the first LP. It is easy to check that this gives a feasible solution. In the m-th LP use the x values of the solution to the previous LP and $\gamma_m = 0$ as the initial solution. This solution is feasible as all but the last constraint are identical to the previous LP and, thus, the x-values of the previous solution fulfill them. For the last constraint, note that the right side of the inequality equals the next-to-last constraint. As all utility values are non-negative, summing over a larger set G on the left side only increases the value of the left side in comparison to the value of the next-to-last constraint. Thus, the last constraint is fulfilled as well for $\gamma_m = 0$.

(2) Given a vector of x-values and a vector of minimum utilities γ_i the goal of a separation oracle is to decide whether these values fulfill the LP and, if they do not, find a constraint that is violated by them. The time of the separation oracle dominates the running time per iteration of the ellipsoid algorithm. Thus, it suffices to give a polynomial-time separation oracle. We present our separation oracle in Algorithm 2. It first checks whether all x-values fall into the correct range and add up to k. Then it computes the utility y_j of each subgroup G_j and sorts them in non-decreasing order of y-value. Instead of checking all $\binom{m}{l}$ constraints for each set of l subgroups, it uses the following observation: it suffices to check that, for each l, the sum of the utilities of the l groups with *smallest* utilities is at least $\sum_{s=1}^{l} \gamma_s$. The reason is that every other set of l subgroups must have cumulative utility at least as large. If, however, the set of l subgroups with minimum utility does not have high enough cumulative utility, then a violating constraint has been found.

Summing up utilities across n candidates takes $O(n)$ time, sorting the resulting utility vector y takes $O(m \log m)$ time. In total, this separation oracle checks if all the constraints are satisfied in $O(m \log m + n)$ time.

Algorithm 2 SEPARATION ORACLE. Checking if a constraint has been violated by a given solution x and γ.

Input: $v \in \mathbb{R}_{\geq 0}^{n \times m}$, values of each group for each candidate, $\{x_1, \ldots, x_n\}$ candidate solution, $\{\gamma_1, \ldots, \gamma_l\}$ minimum utilities for the l-th LP
Output: {TRUE or a violated constraint}
$S \leftarrow 0$;
for $i = 1, \ldots, n$ **do**
  if $x_i > 1$ *or* $x_i < 0$ **then**
    return $\{0 \leq x_i \leq 1\}$
  $S \leftarrow S + x_i$;
if $S \neq k$ **then**
  return $\{\sum_{i=1}^{n} x_i \leq k\}$
$y_j \leftarrow \sum_{i=1}^{n} v_{ij} x_i \quad \forall j = 1, \ldots, m$;
$\tilde{y} \leftarrow \text{SORT}(y)$;
$U_{min} \leftarrow 0$;
for $l = 1, \ldots, m$ **do**
  $U_{min} \leftarrow U_{min} + \tilde{y}_l$;
  if $U_{min} < \sum_{s=1}^{l} \gamma_s$ **then**
    return FALSE as this constraint does not hold:
    $\{\sum_{i=1}^{n} \sum_{G_j \in S} v_{ij} x_i \geq \sum_{s=1}^{l} \gamma_s \quad \forall S \subseteq \mathcal{G} \; s.t. \; |S| = l\}$
return TRUE

For the ellipsoid method, we are guaranteed convergence in k steps where $k \leq 2n^2 \log(\frac{R}{r})$ where R is the initial radius and r is the final radius of the feasible region [4]. For our feasibility region, R is exponential with respect to the input size (i.e. $O(2^n)$) which means $\log(\frac{R}{r})$ is linear with respect to n. Since the separation oracle and centroid method at each step runs in polynomial time and there are at most $\tilde{O}(n^2)$ steps, Algorithm 1 also runs in polynomial time. ◀

B Approximations in the Presence of Noise

So far, we have considered approximate leximax solutions with the assumption that the utilities used to calculate these solutions are known to be correct. However, a natural question is how such approximations behave if the reported utilities contain some small amount of noise.

In the case of ϵ-tradeoff leximax solutions, assuming a small amount of additive noise for each utility has the potential for resulting in solutions that do not satisfy tradeoff guarantees. In particular, noise that is solution-specific can cause individual solutions to be "kicked out" of the recursively defined sets, even though all solutions near them are included. We demonstrate this behavior in the following example:

▶ **Example 17** (ϵ-tradeoff leximax solutions are not robust to noise.)**.** We consider a setting in which we have two groups, $\mathcal{G} = \{G_1, G_2\}$ and three potential solutions $\mathcal{S} = \{S_1, S_2, S_3\}$. The utilities each group derives are defined as $u(S_i, G_j) = U_{ij}$ where U is defined as follows (assume $\epsilon << 0.1$):

$$U = \begin{bmatrix} 0.1 & 0.2 \\ 0.1 + \epsilon/100 & 0.8 \\ 0.1 + \epsilon & 0.2 \end{bmatrix}.$$

Furthermore, assume we have a slightly noisy version of utilities in which $u(S_2, G_1)$ changes from $0.1 + \epsilon/100$ to $0.1 - \epsilon/100$. Figure 5 provides a visual representation of this instance, where the noisy verison of S_2 is shown in red.

In the non-noisy version, S_1 can never be considered ϵ-tradeoff leximax because S_2 does much better than S_1 on G_2, and is still above S_1 on G_1.

However, in the noisy version, which introduces only a tiny amount of noise ($\epsilon/50$), much smaller than the allowed approximation threshold (ϵ), results in a setting where S_1 can be considered ϵ-tradeoff leximax.

By making the distance between S_2 and S_1 arbitrarily small, we can construct examples where even when the amount of noise is negligible compared to the allowed approximation factor, S_1 can still potentially be incorrectly classified as ϵ-tradeoff leximax.

We note that we can define a stricter notion of tradeoff approximation that guarantees a solution will be ϵ-tradeoff leximax even if calculated with noisy utilities, but for the same reasons as demonstrated in Example 8, such solutions may not always exist, making it difficult to find solutions that are guaranteed to be ϵ-tradeoff leximax in a noisy setting.

▶ **Lemma 18.** *Recall the notion of (ϵ_1, ϵ_2)-significant tradeoff leximax as presented in Definition 9. Any $(\epsilon - 2\delta, 2\delta)$-significant tradeoff leximax solution when calculated using noisy utilities within an additive δ of their true values is guaranteed to be ϵ-tradeoff leximax with respect to the true utilities.*

Having considered how noise may affect ϵ-tradeoff leximax approximations, we now turn to ϵ-significant recursive leximax approximations. Here, we find that ϵ-significant recursive leximax solutions are somewhat robust to noise, in that they satisfy a slightly relaxed definition of significance.

First, we note that in Example 17, S_1 is also ϵ-significant recursive leximax in the noisy setting, but not in the non-noisy setting, and so this example also demonstrates how the standard definition of ϵ-significant recursive leximax may not be robust to noise. However, we can offer the following guarantee with respect to a modified notion:

▶ **Lemma 19.** *Say that a solution S is (α_1, α_2)-significant recursive leximax if there exists some choice of slack $\vec{\beta} = (\beta_1, ..., \beta_m)$ with $\beta_i : \mathcal{S} \to [\alpha_1, \alpha_2]$ such that $S \in \mathcal{S}_m^\beta$, where $\mathcal{S}_0^\beta = \mathcal{S}$ and*

$$\mathcal{S}_i^\beta = \{S \in \mathcal{S}_{i-1}^\beta : u(S, G_{[i]}) \geq \max_{S' \in \mathcal{S}_{i-1}^\beta} u(S', G_{[i]}) - \beta_i(S)\}.$$

Then, any ϵ-tradeoff leximax solution calculated in the presence of δ additive noise is guaranteed to be $(\epsilon - 2d, \epsilon + 2d)$-significant recursive leximax.

Thus, we conclude that while noisy ϵ-significant recursive leximax solutions are not guaranteed to be ϵ-significant recursive leximax with respect to the true utilities, they will still satisfy a slightly relaxed notion of significance that allows for slack to vary within an interval of size 4δ around the constant ϵ slack used in standard significance. When δ is tiny compared to ϵ, this is only a tiny change in the allowed slack values.

C Rounding Distributions Over Candidates

Once we obtain a distribution over cohort candidates from Algorithm 1, we can sample each individual i with probability x_i independently. The total size of the committee follows a Poisson Binomial distribution which will be size-k in expectation where $k = \sum_{i=1}^{n} x_i$ according to our constraints.

If a committee of size k is a hard constraint, we can instead take a rounding approach similar to previous work in cohort selection [2]. For finite samples in our cohort selection setting, we can employ a dependent rounding scheme that guarantees that the utilities for each subgroup is leximax in expectation while the size of the cohort is exactly k [28].

The rounding scheme described in [28] results in a distribution over cohorts of size exactly k such that the marginal inclusion probability for each potential cohort member is still satisfied, giving us the leximin utility values in expectation when the utility function is linear over cohort members. The scheme has the added benefit that the events corresponding to the inclusion/exclusion of each cohort member are negatively correlated. Because Chernoff bounds such as the one used in our proof of Lemma 15 have been shown to also hold in settings where random variables are not independent but are negatively correlated (See [11], Theorem 1.10.24), our concentration guarantees also apply to solutions outputted by the rounding scheme.

D Hardness of Integer Solutions

Although the focus of this work is providing distributions over candidates and cohorts, we also touch briefly on the problem of finding integer leximax cohorts. An exact integer leximax solution removes the randomness inherent in rounding from a distributional solution. However, we show such an integer solution is NP-hard to find. Moreover, the weaker maxmin version of the problem (see below) is NP-hard to compute.

Given a set of candidates $\mathcal{C} = \{c_1, \ldots, c_n\}$, a set of groups $\mathcal{G} = \{G_1, \ldots, G_m\}$, and the values $v \in \mathbb{R}_{\geq 0}^{n \times m}$ of each group for each candidate such that the utility of a group for a cohort is its average value over the cohort's candidates, the integer leximax cohort selection problem is:

$$\begin{array}{ll} \text{maximize} & \gamma_1, \ldots, \gamma_m \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & x_i \in \{0, 1\} \\ & \sum_{i=1}^{n} \sum_{G_j \in G} v_{ij} x_i \geq \sum_{s=1}^{l} \gamma_s \quad \forall G \subseteq \mathcal{G} \text{ s.t. } |G| = l, l = 1, \ldots, m. \end{array}$$

The simpler integer *maxmin cohort selection problem with cardinality k* determines a set of candidates defined by x_i's such that the minimum utility of any group is maximized:

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & \sum_{i=1}^{n} x_i = k \\ & x_i \in \{0, 1\} \\ & \sum_{i=1}^{n} x_i v_{ij} \geq \gamma \quad \forall j = 1, \ldots, m. \end{array}$$

Next we show the hardness of the maximin cohort selection problem and even of the following *integer ϵ-approximate maxmin cohort selection problem with cardinality k*, where $0 \leq \epsilon$ is a constant: Determine a set of candidates defined by x_i's such that the minimum utility of any group is within an additive error of ϵ of γ, the maximum minimum utility possible.

▶ **Lemma 20.** *For $\epsilon < 0.5$ the integer ϵ-approximate maxmin cohort selection problem is NP-hard. It is also NP-hard to determine the number of groups with non-minimum utility to within a factor of $(e-1)/e + o(1)$.*

E Figures and Diagrams

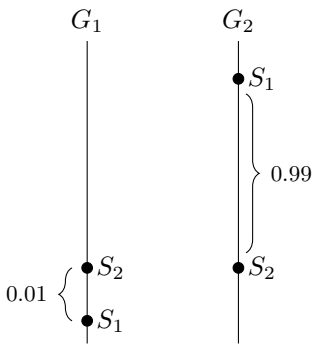

Figure 2 Visual representation of the setting in Example 5 showing how exact leximax solutions are very sensitive to small changes in utility for less-well-off groups.

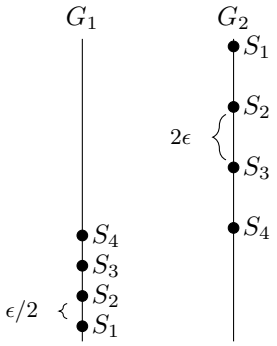

Figure 3 Visual representation of the setting in Example 8 demonstrating how no solutions may exist under small alterations to the definition of ϵ-tradeoff leximax.

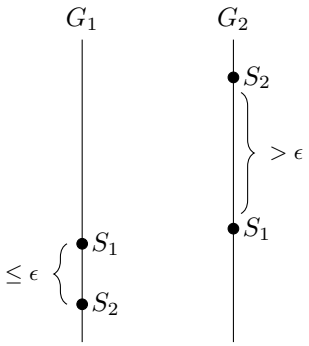

Figure 4 Visual representation of the setting in Example 13 demonstrating how Definition 12 identifies significantly leximax solutions.

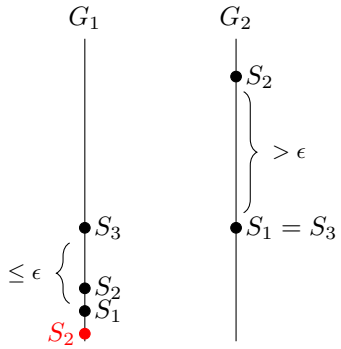

Figure 5 Visual representation of the setting in Example 17 showing that when computed in the presence of noise, ϵ-tradeoff leximax solutions may break down. The noisy version consists of updating S_2 to the location highlighted in red.

On Classification of Strategic Agents Who Can Both Game and Improve

Saba Ahmadi
Toyota Technological Institute at Chicago, IL, USA

Hedyeh Beyhaghi
Carnegie Mellon University, Pittsburgh, PA, USA

Avrim Blum
Toyota Technological Institute at Chicago, IL, USA

Keziah Naggita
Toyota Technological Institute at Chicago, IL, USA

Abstract

In this work, we consider classification of agents who can both game and improve. For example, people wishing to get a loan may be able to take some actions that increase their perceived credit-worthiness and others that also increase their true credit-worthiness. A decision-maker would like to define a classification rule with few false-positives (does not give out many bad loans) while yielding many true positives (giving out many good loans), which includes encouraging agents to improve to become true positives if possible. We consider two models for this problem, a general discrete model and a linear model, and prove algorithmic, learning, and hardness results for each.

For the general discrete model, we give an efficient algorithm for the problem of maximizing the number of true positives subject to no false positives, and show how to extend this to a partial-information learning setting. We also show hardness for the problem of maximizing the number of true positives subject to a nonzero bound on the number of false positives, and that this hardness holds even for a finite-point version of our linear model. We also show that maximizing the number of true positives subject to no false positive is NP-hard in our full linear model. We additionally provide an algorithm that determines whether there exists a linear classifier that classifies all agents accurately and causes all improvable agents to become qualified, and give additional results for low-dimensional data.

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design; Theory of computation → Sample complexity and generalization bounds

Keywords and phrases Strategic Classification, Social Welfare, Learning

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.3

Related Version *Extended Version*: `https://arxiv.org/pdf/2203.00124.pdf` [2]

Funding This work was supported in part by the National Science Foundation under grants CCF-1733556 and CCF-1815011 and by the Simons Foundation under the Simons Collaboration on the Theory of Algorithmic Fairness.

1 Introduction

Consider a bank offering loans. Based on observable information about applicants, it must decide which of them are loan-worthy and which are not. For example, it might compute a credit score based on some (perhaps linear) function of observable features and then compare the result to a cutoff value. So far, this looks like a standard binary classification problem. However, there is an additional wrinkle: individuals have agency and may be able to modify their observable features somewhat if it will help them get approved for a loan. This wrinkle brings both challenges and opportunities. A challenge is that some of these

actions may involve "gaming" the system: performing activities that do not affect their true loan-worthiness such as changing how they spend on different credit cards. An opportunity is that other actions, such as taking a money-management course, may truly help them become more loan-worthy, increasing the number of good loans the bank can give out. How can the bank best set its loan criteria in such settings to maximize the number of loans given out subject to not giving loans to unqualified applicants?

Or, consider a school that would like to prepare students for the workforce. There are many different career paths a student might take, so the school would like to have multiple different criteria for graduation (multiple tracks or majors) such that satisfying any one of them will earn the student a diploma. Imagine there is a limited set of options the school can choose from, and once the school chooses some subset of them as criteria, every student selects the easiest of those criteria to fulfill (or none, if all are too hard) and then may or may not become truly qualified for the workforce, depending perhaps on the extent to which satisfying that criterion involved gaming versus true improvement. How can the school best select criteria to maximize the number of students who become truly qualified for the workforce while minimizing the number of diplomas given to unqualified students?

In this work we consider algorithmic and learning-theoretic formulations of such scenarios, where a binary classification must be made in the presence of both gaming and improvement actions with a goal of maximizing true-positive predictions while keeping false-positives to a minimum. Specifically, we consider the following two formulations (given in more detail in Section 2).

General Discrete Model: In this formulation, we are given a weighted, colored bipartite graph with n nodes on the left representing agents, and m nodes on the right representing distinct possible ways agents could be considered *qualified* for the prize at hand (the loan, the diploma, etc.). For example, the nodes on the right could represent different possible definitions of "credit-worthy" or could represent different bundles of activities sufficient to receive a diploma. Each edge has both a *weight* representing the amount of effort the agent would need to achieve the given qualification and a *color* blue or red indicating whether the agent would indeed be truly qualified or not (respectively) if it did so. The goal of the classifier is to select a subset $\mathcal{P}^{\text{final}}$ of points on the right such that if each agent in the neighborhood of $\mathcal{P}^{\text{final}}$ takes its least-cost edge into $\mathcal{P}^{\text{final}}$, then a large number of blue edges and very few red edges are taken (many good loans and few bad loans are given out); more specific objectives will be detailed in Section 3.

In the learning-theoretic version of this problem, the left-hand-side of the graph is replaced with a probability distribution $\mathcal{D}$ over nodes (where a node is given by its neighborhood and the weights and colors of its edges). We have sampling access to $\mathcal{D}$ and our goal is to find a subset $\mathcal{P}^{\text{final}}$ of points on the right-hand-side with good performance under $\mathcal{D}$. In a partial-information version, when we sample a point from $\mathcal{D}$ we do not get to observe its edges, only where the agent goes to and whether it was qualified. That is, learning proceeds in rounds, where in each round we choose a subset $\mathcal{P}'$ of points on the right, and then for a random draw $x \sim \mathcal{D}$ we observe what point $p \in \mathcal{P}'$ (if any) was selected and the color of the edge taken.

Linear Model: In this formulation, we assume agents are points $\mathbf{x} \in \mathbb{R}^d$ (they have d real-valued features) and there is a linear separator $f^* : \mathbf{a}^*\mathbf{x} \geq b^*$ with non-negative weights that separates the truly qualified individuals from the unqualified ones. Agents have the ability to increase their jth feature at cost $\mathbf{c}[j]$ (decreasing is free) and receive value 1 for being classified as positive. However, only some features correspond to true improvement and others involve just gaming. That is, if an agent begins at $\mathbf{x}^{\text{init}}$ and moves to a

point $\mathbf{x}^{\text{perc}}$, their true qualification is not $f^*(\mathbf{x}^{\text{perc}})$ but rather $f^*(\mathbf{x}^{\text{true}})$, where $\mathbf{x}^{\text{true}}$ agrees with $\mathbf{x}^{\text{init}}$ in the gaming directions and with $\mathbf{x}^{\text{perc}}$ in the improvement directions. Movement costs and which features are improvement versus gaming are assumed to be the same for all agents. The goal is to find a classifier that produces a large number of true positives and few false positives. Note that using f^* itself will be optimal if the coordinate j maximizing $\mathbf{a}^*[j]/\mathbf{c}[j]$ (having the most "bang per buck") is an improvement direction, so the interesting case is when this is a gaming direction. Also note that shifting f^* in this direction (adding $\mathbf{a}^*[j]/\mathbf{c}[j]$ to b^*) will be a perfect classifier but may not be optimal because it does not take advantage of the ability to encourage agents to improve. We consider settings where (a) the mechanism designer must use a linear classifier, (b) arbitrary classifiers are allowed, and (c) a polynomial-sized set $\mathcal{P}$ of "target points" is given and the mechanism designer must select some subset $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$ as its classifier – this is a special case of our General Discrete Model.

In this work, we consider both models. We give an efficient algorithm for the general discrete model for the problem of maximizing the number of blue edges taken subject to no red edges taken (maximizing the number of good loans given out subject to no bad loans) and show how to extend this to the partial-information learning setting. We also show hardness for the problem of maximizing the number of blue edges subject to a nonzero bound on the number of red edges, and show that this hardness holds even for the simplest finite-point linear model. Furthermore, we show the problem of maximizing the number of true positives subject to no false positives is NP-hard in the linear model when we are not given a polynomial-sized set of target points. We additionally give algorithms for the linear model. We provide an algorithm that determines whether there exists a linear classifier which classifies all agents accurately and causes all improvable agents to become qualified. In the special two-dimensional case, we design a linear classifier maximizing the number of true positives minus false positives; and a general (not necessarily linear) classifier that maximizes true positives subject to no false positives.

1.1 Related Work

There is an exciting and growing literature on decision-making in the presence of strategic agents. Much of this work considers agents whose actions are only gaming and do not change their true label (see [11, 7, 13, 16, 1, 6, 9, 5] among others) but researchers have also been investigating mechanism design in the presence of agents who can both game and improve [14, 12, 3, 18, 15, 10, 4, 17].

Kleinberg and Raghavan [14] consider a single agent with a variety of gaming and improvement actions available, that are then converted into observable features through an effort-conversion matrix. They then examine mechanisms for incentivizing desired action vectors, showing among other things that any vector that can be incentivized by a monotone mechanism can also be incentivized by a linear mechanism. Harris et al. [12] consider a multi-round version of the Kleinberg and Raghavan [14] model in which true improvements carry over to future rounds whereas gaming effort do not; they show that in this model, the principal (the decision-maker) can incentivize the agent to produce a greater range of desirable behaviors.

Alon et al. [3] consider a multi-agent extension of the Kleinberg and Raghavan [14] model, where agents all begin at the same place (the origin) but each have their own effort-conversion matrix. The goal of the designer is to choose an evaluation mechanism – mapping observable features to payoffs – that encourages all agents to take *admissible* actions, assuming that

agents will maximize payoff subject to budget constraints. They specifically consider the case (1) that there is a single admissible action vector, and (2) that individual actions are either improvement or gaming actions and no agent should take a gaming action. Among other results they show that unlike in [14], nonlinear evaluation mechanisms can now be more powerful than linear ones; they also analyze the complexity of a variety of associated optimization problems. We can think of our setting to some extent in this language by viewing any action that makes an agent truly qualified as "admissible" (and specifically the blue edges in our general discrete model). However, two key distinctions are (1) in our setting we can only give the loan/diploma or not – we do not have the flexibility to choose arbitrary payoffs, and (2) we assume agents may begin at different starting locations (but have the same costs for movement in our linear model).

Xiao et al. [18] define a problem they call the *Multiple Agents Contract Problem* which is very similar to our General Discrete Model, except instead of binary (red/blue) colors, the edges have different values to the principal, and instead of producing a classification, the principal can assign an arbitrary payment profile to the right-hand-side nodes. They prove that maximizing payoff to the principal is NP-hard, and give an algorithm for a case of related agents in which there is a certain strict ordering among agents and costs.

Shavit et al. [17], building on Miller et al. [15], consider the goal of getting agents to improve without loss of predictive accuracy. As in our setting, they assume agents begin a different starting locations, and then modify their profiles from there, and they also consider a learning formulation. However, their focus is on a regression model in which agents' payoffs are an inner product of their observable features with a decision vector; this means that the incentives are basically the same no matter what the initial location of an agent is. In contrast, in our binary classification setting, even in the linear model the effect of a proposed classifier on an agent may depend greatly (and in a non-convex manner) on the initial location of the agent. Bechavod et al. [4] also consider a linear regression learning setting: agents arrive one at a time iid from a fixed distribution and then modify their state by changing a single variable based on the current regression vector. As in our linear model, some directions are improvement and some are gaming. They consider a limited feedback setting where the learner sees only the dot-product of the agent's true position with the true regression function, plus noise, and the learner's goal is to recover the true regression function.

Haghtalab et al. [10] consider a similar setting to ours in which there are improvement and gaming actions, and the designer is limited to binary classification, where agents receive value 1 for being classified as positive. Among other results, they give approximation algorithms for the goal of maximizing the total amount of true improvement that occurs when the allowed mechanisms are linear separators and agents have ℓ_2 movement costs. In contrast, our goal is to maximize true positive classifications while minimizing false positives, and in the linear case our movement cost assumptions are somewhat different.

Organization of the Paper

Section 2 introduces the general discrete model and linear model more formally. In Section 3, we give an efficient algorithm for the problem of maximizing the number of true positives subject to no false positives in the general discrete model, and provide hardness results for the problem of maximizing the number of true positives subject to a nonzero bound on false positives (in either the general discrete model or the linear model when arbitrary classifiers are allowed) and hardness for the problem of maximizing the number of true positives subject to no false positives in the linear model when arbitrary classifiers are allowed. In Section 4, we consider a learning-theoretic version of the problem of maximizing true positives subject

to no false positives, and provide efficient learning algorithms as well as upper and lower bounds on the number of samples needed. In Section 5, we focus on the linear model and provide algorithms specific to this setting. We provide an algorithm that determines whether there exists a linear classifier which classifies all agents accurately and causes all improvable agents to become qualified. In the special two-dimensional case, we design a general (not necessarily linear) classifier that maximizes true positives subject to no false positives. In the full version of this work, we show how to provide a linear classifier maximizing the number of true positives minus false positives in the two-dimensional case.

2 Model

We study a binary classification problem. As the mechanism designer or classifier, we would like to maximize the number of agents we correctly classify as positive (true positives), and minimize the number of unqualified agents we misclassify as positive (false positives).

Agents are assumed to be utility maximizers and wish to be classified as positive. Each agent $i \in \{1, \ldots, n\}$ has a set of actions it can perform, and it will choose the cheapest of these that causes it to be classified as positive if that cost is less than its value on receiving a positive classification. We use $\mathcal{Q}$ to denote the set of truly qualified agents. If an agent is initially not qualified (not in $\mathcal{Q}$), some of its actions may cause it to become truly qualified, whereas others may not. However, the classifier cannot see which action was taken, only the observable result of that action. Therefore, the challenge of the mechanism designer is to determine which observable results to classify as positive to maximize correct positive classifications while minimizing false positives.

2.1 General Discrete Model

In this model, we assume that as a mechanism designer we are given a polynomial-sized set $\mathcal{P}$ of criteria we may select from (e.g., graduation criteria or criteria for being approved for a loan), and are limited to choosing some subset $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$ as the criteria we will use. We then will classify as positive any agent that meets any one of these criteria, and as negative any agent who does not. Specifically, we are given a weighted, colored bipartite graph with the n agents on the left and the set $\mathcal{P}$ of criteria on the right. Edge (i, j) corresponds to agent i taking an action to satisfy criteria j and is colored blue or red depending on whether that action would make the agent truly qualified or not, respectively. Each edge also has a weight representing its cost to that agent, and only actions whose costs are less than the value to the agent of being classified as positive are shown. Given a set $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$ chosen by the mechanism designer, each agent in the neighborhood of $\mathcal{P}^{\text{final}}$ will choose its cheapest edge into $\mathcal{P}^{\text{final}}$ as the action it will take, and will be classified as positive by the mechanism; agents not in the neighborhood of $\mathcal{P}^{\text{final}}$ will be classified as negative.

We also consider a learning-theoretic version of this problem, where the left-hand-side of the graph is replaced with a probability distribution $\mathcal{D}$ over nodes. We have sampling access to $\mathcal{D}$ and our goal is to find a subset $\mathcal{P}^{\text{final}}$ of points on the right-hand-side with good performance under $\mathcal{D}$. In a partial-information (bandit-style) version, when we sample a point from $\mathcal{D}$ we do not get to observe its edges, only where it goes to and whether it was qualified. That is, learning proceeds in rounds, where in each round we choose a subset $\mathcal{P}'$ of points on the right, and then for a random draw $x \sim \mathcal{D}$ we observe what point $p \in \mathcal{P}'$ (if any) was selected and the color of the edge taken.

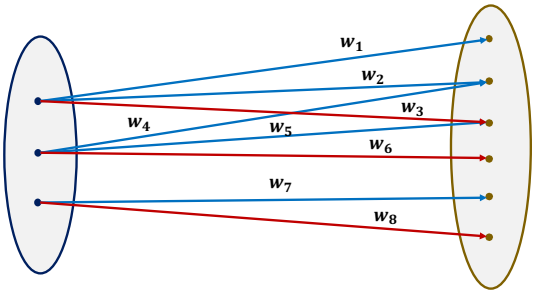

Figure 1 *Points on the left are the agents, and those on the right are the set $\mathcal{P}$ of possible criteria; w_i is the cost of satisfying the criterion. A red edge means the agent taking that action would not truly be qualified. A blue edge means that the agent taking that action would be qualified.*

2.2 Linear Model

In the linear model, agents have d real-valued features. Each agent i begins at an initial point $\mathbf{x}_i^{\text{init}} \in \mathbb{R}^d$, and there is assumed to be a linear threshold function $f^* : \mathbf{a}^*\mathbf{x} \geq b^*$ with non-negative weights that separates the truly qualified individuals from the unqualified ones. Agents have the ability to increase their jth feature at cost $\mathbf{c}[j]$ (decreasing is free) and receive value 1 for being classified as positive. However, only some features correspond to true improvement and others involve just gaming. That is, if an agent begins at $\mathbf{x}^{\text{init}}$ and moves to a point $\mathbf{x}^{\text{perc}}$, their true qualification is not $f^*(\mathbf{x}^{\text{perc}})$ but rather $f^*(\mathbf{x}^{\text{true}})$, where $\mathbf{x}^{\text{true}}$ agrees with $\mathbf{x}^{\text{init}}$ in the gaming directions and with $\mathbf{x}^{\text{perc}}$ in the improvement directions. On the other hand, the classification rule can only be based only on $\mathbf{x}^{\text{perc}}$ and not $\mathbf{x}^{\text{true}}$ (or $\mathbf{x}^{\text{init}}$). Movement costs and which features are improvement versus gaming are assumed to be the same for all agents. So, for any agent i, $cost(\mathbf{x}_i^{\text{init}}, \mathbf{x}_i^{\text{perc}}) = \sum_{j=1}^{d} \mathbf{c}[j]\,(\mathbf{x}_i^{\text{perc}}[j] - \mathbf{x}_i^{\text{init}}[j])^+$, where $x^+ = \max\{x, 0\}$ and $\mathbf{c}[j]$ is the cost per unit of movement in the positive direction of dimension j. An example is given in Figure 2.

We consider settings where (a) the mechanism designer must use a linear classifier (a linear threshold function), (b) arbitrary classifiers are allowed, and (c) a polynomial-sized set $\mathcal{P}$ of "target points" is given and the mechanism designer must select some subset $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$ as its classifier. Notice that this last case is a special case of the general discrete model because given each initial state $\mathbf{x}_i^{\text{init}}$, we can compute the costs to move to each $p \in \mathcal{P}$ and whether doing so will make the agent truly qualified, to produce the desired weighted, colored bipartite graph.

3 Algorithmic and Hardness Results

In this section we first provide an algorithm for the problem of maximizing the number of true positives subject to no false positives in the general discrete model. Then, we provide hardness results for the problem of maximizing the number of true positives subject to a nonzero bound on false positives (in either the general discrete model or the linear model when arbitrary classifiers are allowed) and hardness for the problem of maximizing the number of true positives subject to no false positives in the linear model when arbitrary classifiers are allowed. Later in Section 4 we extend our algorithmic results to the learning model and in Section 5 we give algorithms for learning linear classifiers in the linear model.

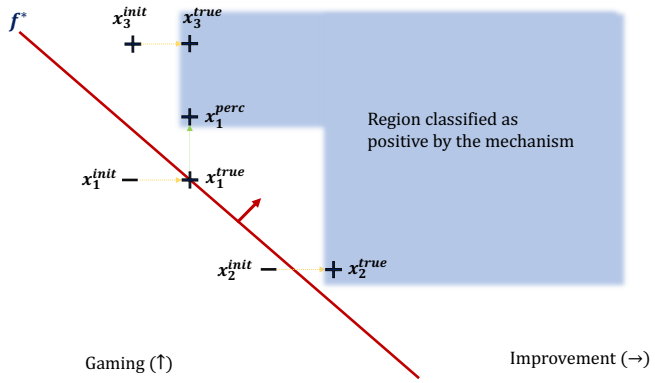

Figure 2 *An example of the linear model (the horizontal axis is an improvement direction and the vertical axis is a gaming direction) with a mechanism using a non-linear classifier. There are three agents, two of whom are initially not qualified. All three become qualified and are correctly classified as positive by the mechanism.*

3.1 Maximize True Positives Subject to No False Positives

The main result of this section is an algorithm that given a weighted, colored bipartite graph $\mathcal{G}$ with agents, $\mathcal{X}$, on the left and potential criteria, $\mathcal{P}$, on the right, finds $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$ such that using $\mathcal{P}^{\text{final}}$ as the criteria maximizes the number of agents taking a blue edge (true positive) subject to no agent taking a red edge (false positive). We call the agents that take a blue edge *improving agents* and the agents taking a red edge *gaming agents*. The algorithm, although simple in structure, satisfies strong properties noted afterwards; and serves as the building block of the learning algorithms in Section 4. Furthermore, as shown in the following subsection, natural generalizations of the objective function make the problem computationally hard. Therefore, the algorithm together with the hardness results tightly characterize the settings for which there is an efficient algorithm, or the problem is NP-hard.

Overview of Algorithm 1. The algorithm takes in a weighted, colored bipartite graph $\mathcal{G} = (\mathcal{X} \cup \mathcal{P}, E)$ and outputs $\mathcal{P}^{\text{final}}$, a subset of $\mathcal{P}$ that specifies the final criteria. Initially, $\mathcal{P}^{\text{final}}$ is set to $\mathcal{P}$. The algorithm proceeds in rounds. In each round, it visits all the nodes (agents) in $\mathcal{X}$ to determine whether there is an agent who takes a red edge to its lowest cost neighbor $p \in \mathcal{P}^{\text{final}}$. If there is such a gaming agent, its corresponding criteria, p, is removed from $\mathcal{P}^{\text{final}}$. These rounds continue until there is no gaming agent and therefore no removal of criteria in a single round, or the current set of criteria is empty.

▶ **Proposition 1.** *Algorithm 1 has running time of $O(|\mathcal{P}|n)$.*

Proof. Proof in Appendix A. ◀

▶ **Theorem 2.** *Algorithm 1 finds the set of criteria, $\mathcal{P}^{final}$, that maximizes the number of true positives subject to no false positive.*

Proof. Proof in Appendix A. ◀

Algorithm 1 satisfies the following strong properties.

(a) *point-wise optimality*: For any agent i, if there exists a solution in which i takes a blue edge and no agent takes a red edge, then the algorithm finds such a solution.

Algorithm 1 Maximize true positives subject to no false positives.

Input : A bipartite graph $\mathcal{G} = (\mathcal{X} \cup \mathcal{P}, E)$ with edge weights w_e. Outgoing edges assumed sorted by weight. Red edges $E_R \subseteq E$. Blue edges $E_B \subseteq E$.
Output : $\mathcal{P}^{\text{final}}$

$\mathcal{P}^{\text{final}} \leftarrow \mathcal{P}$ // Initialization of the set
while $\mathcal{P}^{final} \neq \emptyset$ **do**
  $flag = 0$
  /* Loop through all $x_i \in \mathcal{X}$ */
  for $i = 1, 2, \cdots$ **do**
    Let $e = (x_i, p \in \mathcal{P}^{\text{final}})$ be the outgoing edge from x_i with lowest weight
    if $e \in E_R$ **then**
      $flag = 1$ // at least one agent is gaming
      $\mathcal{P}^{\text{final}} \leftarrow \mathcal{P}^{\text{final}} \setminus \{p\}$
  if *flag is* 0 **then**
    return $\mathcal{P}^{\text{final}}$
return $\emptyset$ // When 0 false positive is not possible

(b) *general for weighted setting*: The algorithm works optimally in the more general setting that each agent has a weight and the objective is to maximize the sum of weights of improving agents subject to the constraint of no gaming agent. This is a direct implication of property a.

(c) *max-min fairness:* Suppose the agents are from different populations and the objective is to maximize the minimum number of agents improving from each population subject to no gaming. By property a, the algorithm satisfies this max-min fairness notion.

(d) *heterogeneous utilities*: The algorithm works optimally in the more general setting that agents have different values for being classified positive.

(e) *minimizing the total cost of improvement*: Since the algorithm only removes $p \in \mathcal{P}$ that causes an agent to game, with $\mathcal{P}^{\text{final}}$ each agent incurs the minimal cost subject to no agent gaming.

▶ Remark 3. The sets of criteria satisfying the no false positive constraint is not downward closed. In other words, a subset of a set of criteria that satisfies the no false positives property does not necessarily satisfy this property.

3.2 Hardness Results

In this part, we prove hardness results for maximizing the number of true positives when the constraints in the previous subsection are relaxed. First, we show that if we relax the no false positives constraint to a bounded number of false positives, the problem becomes NP-hard; moreover, this holds even for the simpler linear model. Then, for the linear model, we show if we are not given a finite set of potential criteria $\mathcal{P}$, it is NP-hard to find criteria that maximize true positives subject to no false positives.

▶ **Theorem 4.** *Given the initial feature vectors of agents $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init} \in \mathbb{R}^d$ and a set $\mathcal{P}$ of potential criteria, the problem of finding a subset $\mathcal{P}^{final} \subseteq \mathcal{P}$ that maximizes the number of true positives subject to at most k false positives is NP-hard.*

Proof sketch. The proof is done by a reduction from the Max-k-Cover problem with n elements where the goal is to choose k sets covering the most elements. For every element e_i in the Max-k-Cover, we consider agent i, and for every set S_j in the Max-k-Cover problem

we consider agent $n+j$ and a target point $\mathbf{p}_j$. The coordinates of the initial points and the target points are set such that agent i corresponding to element e_i can only move to target point $\mathbf{p}_j$ such that $e_i \in S_j$ and become a true positive; moreover, agent $n+j$ corresponding to set S_j can only move to target point $\mathbf{p}_j$ and become a false positive. On the one hand, since including each $\mathbf{p}_j$ in the final set of criteria, $\mathcal{P}^{\text{final}}$, causes exactly one agent to be a false positive, $\mathcal{P}^{\text{final}}$ must contain at most k target points. On the other hand, to maximize the number of true positives a set of k target points that the maximum number of agents can reach to it must be selected. This is equivalent to the Max-k-Cover solution. A formal proof is included in Appendix A. ◀

▶ **Theorem 5.** *Suppose we are given a set of n agents where $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init}$ denote their initial feature vectors. Deciding whether there exists a set of target points $\mathcal{P}^{final} \subseteq \mathbb{R}^d$ for which all the agents become true positives is NP-hard.*

Proof sketch. The proof is done by a reduction from the approximate version of the hitting set problem where given a set of elements, $\mathcal{E} = \{e_1, \ldots, e_n\}$ and a family of sets of elements, $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$, the goal is to find a minimum size set S^* that intersects all S_i. We construct an $n+1$-dimensional space, where the first n dimensions are improvement dimensions and correspond to the n elements, and the last dimension is gaming. We consider two sets of agents. For each S_i, we consider a corresponding agent i; these are the *usual* agents. We also consider agent $m+1$, a *special* agent that does not correspond to any particular set. The construction is such that each agent needs to move $2k$ units along the improvement dimensions to become truly qualified. Further details of the construction can be found in the full proof. The proof includes two directions. (1) If all the agents can become true positives by reaching to a set of target points $\mathcal{P}^{\text{final}} \subseteq \mathbb{R}^d$, then we can construct a hitting set of size at most $2k$; and (2) if it is not possible, then there does not exist a hitting set of size k.

We briefly cover the key ideas in each direction. To show the first direction, suppose all the agents can become true positives when presented with target points $\mathcal{P}^{\text{final}} \subseteq \mathbb{R}^d$. Consider the target point that each agent selects. Using our construction, we show the special agent does not afford to reach to the target points of the usual agents. Also, for each usual agent i, there exists element e_j in their corresponding set such that the target point of the special agent has value more than 1 in coordinate j. In order for the special agent to afford to reach to its target point, the number of improvement coordinates with value at least 1 must be at most $2k$. The elements corresponding to these coordinates constitute a hitting set of size at most $2k$. To prove the reverse direction we argue: if there exists a hitting set S^* of size k, there is a set of target points that encourages all the agents to become true positives. To do so, we construct a set of target points $\mathcal{P}^{\text{final}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{m+1}\}$, using the elements in the hitting set, that when the size of the hitting set is k makes every agent become true positive. A formal proof is included in Appendix A. ◀

The following is a direct corollary of Theorem 5.

▶ **Corollary 6.** *Given the initial feature vectors of agents, $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init} \in \mathbb{R}^d$, finding a set of target points $\mathcal{P}^{final} \subseteq \mathbb{R}^d$ that maximizes the number of true positives subject to no false positives is NP-hard.*

4 Learning Results

In this section we consider a learning-theoretic version of our problem, where the left-hand-side of the graph is replaced with a probability distribution $\mathcal{D}$ over nodes. We have sampling access to $\mathcal{D}$ and our goal is to find a subset $\mathcal{P}^{\text{final}}$ of points on the right-hand-side with good

performance under $\mathcal{D}$. We provide two different algorithmic results and upper bounds on the number of samples for producing a good solution, depending on the information each sample reveals. The first upper bound works for the case where by sampling an agent, its neighborhood (neighboring edges, their colors and weights) is revealed. The second upper bound works in a partial-information (bandit-style) setting, where when we sample a point from $\mathcal{D}$ we do not get to observe its edges, only where it goes to and whether it was qualified. Finally, we provide a lower bound on the necessary number of samples for any algorithm. The lower bound holds even for the simpler linear model.

The following definition is crucial in this section.

▶ **Definition 7** (OPT, performance, and error)**.** *Let OPT be the maximum probability mass of true positives achievable subject to zero false positives. We denote the probability mass of true positives of an algorithm as its* performance *and the probability mass of false positives as its* error*. A hypothesis is desired if it has comparable performance to OPT and small error.*

4.1 Sufficient Number of Samples in the Full Information Setting

The main result of this section is that a number of samples linear in $|\mathcal{P}|$ and $1/\varepsilon$ is sufficient for Algorithm 1 to learn a desired hypothesis with high probability. Specifically, suppose the learner has access to a weighted, colored bipartite graph $\mathcal{G} = (\mathcal{X} \cup \mathcal{P}, E)$, where $\mathcal{X}$ are sampled from $\mathcal{D}$, and $\mathcal{P}$ is the set of the potential criteria. The learner runs Algorithm 1 with the graph as the input and uses the algorithm output, $\mathcal{P}^{\text{final}} \subseteq \mathcal{P}$, as its hypothesis, i.e., after the training phase it classifies any agent with an edge to $\mathcal{P}^{\text{final}}$ as positive and any other agent as negative. We show that a linear number of samples is sufficient so that with high probability, the probability mass of true positives classified by $\mathcal{P}^{\text{final}}$ is close to OPT and the probability mass of false positives is small.

▶ **Theorem 8.** *Consider $\mathcal{P}^{final}$ as the outcome of Algorithm 1 on $\mathcal{G} = (\mathcal{X} \cup \mathcal{P}, E)$, where $\mathcal{X}$ contains samples from $\mathcal{D}$. For any $0 < \varepsilon, \delta \leq 1$, if $|\mathcal{X}| \geq \varepsilon^{-1}(\ln(2)|\mathcal{P}| + \ln(1/\delta))$ then with probability at least $1 - \delta$ the set $\mathcal{P}^{final}$ achieves performance at least $OPT - \varepsilon$ (i.e., at least $OPT - \varepsilon$ probability mass of true positives) subject to at most ε error (ε probability mass of false positives).*

4.2 Sufficient Number of Samples in the Partial Information Setting

In this section we consider a partial information (bandit-style) setting. Similar to before, the learner has access to a sample set $\mathcal{X}$ drawn from D and a set of potential criteria $\mathcal{P}$. However, observing a sample in $\mathcal{X}$ does not reveal its edges, and the learner can only observe the criterion that the sample selects and whether it becomes truly qualified. The main result of this section is an algorithm, Algorithm 2, for this setting and a guarantee on the number of samples sufficient for it to achieve performance at least $\text{OPT} - \varepsilon$ and error at most ε with high probability.

Overview of Algorithm 2. In each iteration, a set of examples of size $\varepsilon^{-1}\ln(|\mathcal{P}|/\delta)$ is sampled. After agents select points in $\mathcal{P}$ (if any), we observe the points selected and whether they became truly qualified (in a real-world application, one can think of performing a test to check if each agent is truly qualified). If some agent does not become truly qualified (fails the test), the algorithm deletes the point they have selected. If a set $\mathcal{P}^{\text{final}}$, survives for $\varepsilon^{-1}\ln(|\mathcal{P}|/\delta)$ subsequent examples, the algorithm terminates and returns $\mathcal{P}^{\text{final}}$ as the

Algorithm 2 Learning a high performance low error $\mathcal{P}^{\text{final}}$ in partial-information setting.

Input : $\mathcal{P}$
Output : $\mathcal{P}^{final}$
$\mathcal{P}^{\text{final}} \leftarrow \mathcal{P}$;
while $\mathcal{P}^{final} \neq \emptyset$ **do**
  Sample $\mathcal{X} \sim \mathcal{D}$ of size $\frac{1}{\varepsilon} \ln \frac{|\mathcal{P}|}{\delta}$;
  if $\exists x \in \mathcal{X}$ *such that* x *takes a red edge to* $p \in \mathcal{P}^{final}$ **then**
    $\mathcal{P}^{\text{final}} \leftarrow \mathcal{P}^{\text{final}} \setminus \{p\}$;
    continue;
  `/* if no one from` $\mathcal{X}$ `takes a red edge: */`
  return $\mathcal{P}^{\text{final}}$;
return $\emptyset$;

the final set of criteria of the algorithm. Since the number of false positives (agents taking red edges) is bounded by $|\mathcal{P}|$, the algorithm will terminate after at most $\varepsilon^{-1}|\mathcal{P}| \ln(|\mathcal{P}|/\delta)$ samples.

The following theorem proves that with a high probability, Algorithm 2 outputs $\mathcal{P}^{\text{final}}$ with a high performance and a low error.

▶ **Theorem 9.** *For any* $0 < \varepsilon, \delta \leq 1$*, Algorithm 2 by using at most* $\varepsilon^{-1}|\mathcal{P}| \ln(|\mathcal{P}|/\delta)$ *total samples outputs a set of criteria* $\mathcal{P}^{final}$ *that with probability at least* $1-\delta$ *achieves performance at least* $OPT - \varepsilon$ *(i.e., at least* $OPT - \varepsilon$ *probability mass of true positives) subject to at most* ε *error (*ε *probability mass of false positives).*

4.3 Necessary Number of Samples

The main result of this section is a lower bound on the necessary number of samples for learning a desired hypothesis. The lower bound provided holds even for the simpler linear model. To restate the setup, suppose the learner has access to a set of initial positions of agents $\mathcal{X}$ and a set of potential criteria (also called target points in the linear model) $\mathcal{P}$ where $\mathcal{X}$ are sampled from distribution $\mathcal{D}$. We lower-bound the required number of samples for any learning algorithm that with probability at least $1/2$ achieves high performance and low error.

▶ **Theorem 10.** *Any algorithm for PAC learning a set* $\mathcal{P}^{final}$ *that with probability at least* $1/2$ *achieves performance at least* $(3/4) \cdot OPT$ *(i.e., at least* $(3/4) \cdot OPT$ *probability mass of true positives) subject to at most* ε *error (*ε *probability mass of false positives) must use* $\Omega(|\mathcal{P}|/\varepsilon)$ *examples in the worst case.*

5 Algorithmic Results Specific to the Linear Model

The algorithmic results provided so far work in both the general discrete and the linear discrete models. In this section we focus on the linear model and provide algorithmic results for various problems. These algorithms do not follow the greedy structure of the previous algorithms, and use novel technical ideas. First, we consider the problem of designing *linear classifiers*. Section 5.1 provides introductory observations and definitions about linear classifiers. Section 5.2 presents the main result of this section which determines whether there exists a linear classifier that classifies all agents accurately and causes all improvable

agents to become qualified. Then, we shift focus to general (not necessarily linear) classifiers in a two-dimensional space and in Section 5.3 provide an algorithm for maximizing true positives subject to no false positives. In the full version of this work, we provide results for finding a linear classifier that maximizes the number of true positives minus false positives in the two-dimensional case.

5.1 Properties of Linear Classifiers

Before diving into discussion of the algorithmic results, we provide observations about linear classifiers to set the context. We also provide optimal classifiers in special cases.

For the following discussion, consider linear classifier $f^* : \mathbf{a}^*\mathbf{x} \geq b^*$ that separates the truly qualified agents from unqualified agents.

▶ **Observation 11.** *With linear classifier $f : \mathbf{a}\mathbf{x} \geq b$, any utility maximizing agent that achieves non-negative utility by changing their features moves in dimension* $\arg\max_j \mathbf{a}[j]/\mathbf{c}[j]$.

▶ **Definition 12** (movement dimension)**.** *The movement dimension of linear classifier $f : \mathbf{a}\mathbf{x} \geq b$ is the utility maximizing dimension* $\arg\max_j \mathbf{a}[j]/\mathbf{c}[j]$ *discussed in Observation 11. If there are multiple such dimensions the ties are broken in favor of improvement dimensions and then lexicographically.*

▶ **Definition 13** (encourage improvement/gaming)**.** *A classifier encourages improvement if its movement dimension is an improvement dimension. It encourages gaming otherwise.*

▶ **Definition 14** (dim-j improving)**.** *A linear classifier is dim-j improving if it encourages improvement and its movement dimension is along dimension j.*

The following definition captures the set of agents that potentially can improve to become truly qualified.

▶ **Definition 15** (improvement margin, improvable agents)**.** *The improvement margin includes all the agents that can afford (do not have to incur a cost of more than 1) to move in an improvement dimension and become truly qualified. Formally, any initially unqualified agent i, i.e., $\mathbf{a}^*\mathbf{x}_i^{init} < b^*$, that has distance $\leq 1/\mathbf{c}[j]$ along an improvement dimension j to f^* is in the improvement margin.*

▶ **Lemma 16.** *If $f^* : \mathbf{a}^*x \geq b^*$ encourages improvement, the optimal classifier is f^* – among all linear or nonlinear classifiers.*

Proof. f^* classifies initially qualified agents and unqualified unimprovable agents accurately. Also, all the agents in the improvement margin improve, become qualified, and are accurately classified as positive. ◀

▶ **Lemma 17.** *Let j be the movement dimension of classifier f^*. The classifier $g : \mathbf{a}^*\mathbf{x} \geq b^* + \mathbf{a}^*[j]/\mathbf{c}[j]$ classifies all the initially qualified agents as positive and the rest as negative.*

Proof. Initially unqualified agents, $\mathbf{a}^*\mathbf{x}_i^{\text{init}} < b^*$, can move at most $1/\mathbf{c}[j]$ in dimension j which is not enough to reach to g. Therefore, these agents are classified as negative by g. On the other hand, initially qualified agents, $\mathbf{a}^*\mathbf{x}_i^{\text{init}} \geq b^*$, afford to reach to g and receive nonnegative utility. Therefore, they will be classified as positive. ◀

▶ **Corollary 18.** *If all the dimensions are gaming dimensions, $g : \mathbf{a}^*\mathbf{x} \geq b^* + \mathbf{a}^*[j]/\mathbf{c}[j]$ is the optimal classifier, where j is the movement dimension of f^*.*

Proof. If all dimensions are gaming dimensions, there are no improvable agents. Therefore, all agents are either initially qualified or unimprovable and unqualified. By Lemma 17, g classifies all such agents accurately. ◀

By Lemma 17, $g : \mathbf{a}^*\mathbf{x} \geq b^* + \mathbf{a}^*[j]/\mathbf{c}[j]$ may be a "reasonable" solution because it classifies all the initially qualified as positive and does not result in any false positive classifications. However, it misses out on any new true positives resulting from encouraging agents to become qualified. From this point on, we aim to study other classifiers (not necessarily parallel to f^*) with the hope of encouraging other agents to become qualified.

5.2 Linear Classifier for Improvable Agents

In this subsection, we study a problem that takes as input three disjoint subsets of the agents, $\mathcal{S}^{\text{yes}}$, $\mathcal{S}^{\text{no}}$, and $\mathcal{S}^{\text{imp}}$, and outputs a linear classifier (if one exists) that satisfies the following properties.

- **i)** Classifies agent i such that $\mathbf{x}_i^{\text{init}} \in \mathcal{S}^{\text{yes}}$ as positive.
- **ii)** Classifies agent i such that $\mathbf{x}_i^{\text{init}} \in \mathcal{S}^{\text{no}}$ as negative.
- **iii)** Encourages agent i such that $\mathbf{x}_i^{\text{init}} \in \mathcal{S}^{\text{imp}}$ to improve and become truly qualified, i.e., $\mathbf{x}_i^{\text{true}} \in \mathcal{Q}$, and classifies i as positive.

The main result of the section is solving this problem in polynomial time. When $\mathcal{S}^{\text{yes}}$ is the set of initially qualified agents, $\mathcal{S}^{\text{no}}$ is the set of unqualified and unimprovable, and $\mathcal{S}^{\text{imp}}$ is the set of improvable agents, this problem determines whether there exists a linear classifier that classifies $\mathcal{S}^{\text{yes}}$ and $\mathcal{S}^{\text{no}}$ accurately and makes all the improvable agents qualified.

To solve this problem, we divide it into subproblems as following: Does there exist a linear classifier with movement direction in *dimension j* that satisfies properties i, ii, and iii? If the answer is "yes" for some dimension j, then the answer to the main problem is "yes". If the answer is "no" for all $1 \leq j \leq d$, no linear classifier satisfying the three properties exists.

Note that if $\mathcal{S}^{\text{imp}}$ is nonempty, in order to satisfy property iii, dimension j must be an improvement dimension. Therefore, we study the following problem.

▶ **Problem 1.** *Does there exist a dim-j improving classifier (a linear classifier encouraging improvement* in dimension *j) that satisfies properties i, ii, and iii?*

We propose a linear program that solves Problem 1. The following definition and observations illustrate the conditions under which a dim-j improving classifier satisfies each property for agent i.

▶ **Definition 19.** *For a fixed improvement dimension j and classifiers $f^* : \mathbf{a}^*\mathbf{x} \geq b^*$ and $f : \mathbf{a}\mathbf{x} \geq b$, the points $\mathbf{x}_{i,f^*}$, $\mathbf{x}_{i,f}$, $\mathbf{x}_{i,max}$ are defined as follows (depicted in Figure 3.):*

- *$\mathbf{x}_{i,f^*}$ is the projection of $\mathbf{x}_i^{init}$ on the separating hyperplane of classifier f^* along dimension j.*
- *$\mathbf{x}_{i,f}$ is the projection of $\mathbf{x}_i^{init}$ on the separating hyperplane of classifier f along dimension j.*
- *$\mathbf{x}_{i,max}$ is the shifted $\mathbf{x}_i^{init}$ along dimension j by $1/\mathbf{c}[j]$.*

More formally, for all coordinates $k \neq j$, we have $\mathbf{x}_{i,f^*}[k] = \mathbf{x}_{i,f}[k] = \mathbf{x}_{i,max}[k] = \mathbf{x}_i^{init}[k]$. Also, since $\mathbf{a}^*\mathbf{x}_{i,f^*} = b^*$, we have $\mathbf{x}_{i,f^*}[j] = \left(b^* - \sum_{k\neq j} \mathbf{a}^*[k]\mathbf{x}_i^{init}[k]\right)/\mathbf{a}^*[j]$. Similarly, since $\mathbf{a}\mathbf{x}_{i,f} = b$, we have $\mathbf{x}_{i,f}[j] = \left(b - \sum_{k\neq j} \mathbf{a}^*[k]\mathbf{x}_i^{init}[k]\right)/\mathbf{a}[j]$. Finally, $\mathbf{x}_{i,max}[j] = \mathbf{x}_i^{init}[j] + 1/\mathbf{c}[j]$.

▶ **Observation 20.** *A dim-j improving classifier $f : \mathbf{ax} \geq b$ classifies agent i as positive (property i) if $\mathbf{ax}_{i,max} \geq b$. It classifies agent i as negative (property ii) if $\mathbf{ax}_{i,max} < b$.*

▶ **Observation 21.** *Using a dim-j improving classifier f, agent i becomes qualified and is classified as positive (property iii) if and only if $\mathbf{x}_{i,f^*}[j] \leq \mathbf{x}_{i,f}[j] \leq \mathbf{x}_{i,max}[j]$. See Figure 3.*

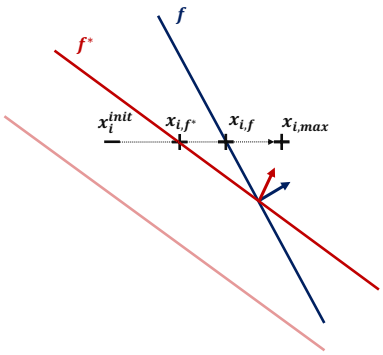

Figure 3 Depicting $\mathbf{x}_i^{\text{init}}, \mathbf{x}_{i,f^*}, \mathbf{x}_{i,f}, \mathbf{x}_{i,\max}$ in Definition 19 and Observation 21. The horizontal axis shows dimension j in the definition.

▶ **Proposition 22.** *The following LP captures Problem 1, where the variables are $\mathbf{a}$ and b.*

$$\frac{\mathbf{a}[k]}{\mathbf{c}[k]} \leq \frac{\mathbf{a}[j]}{\mathbf{c}[j]} \qquad \forall k \neq j \tag{1}$$

$$b \leq \mathbf{ax}_{i,max} \qquad \forall \mathbf{x}_i^{init} \in \mathcal{S}^{yes} \tag{2}$$

$$\mathbf{ax}_{i,max} < b \qquad \forall \mathbf{x}_i^{init} \in \mathcal{S}^{no} \tag{3}$$

$$\mathbf{x}_{i,f^*}[j] \leq \mathbf{x}_{i,f}[j] \qquad \forall \mathbf{x}_i^{init} \in \mathcal{S}^{imp} \tag{4}$$

$$\mathbf{x}_{i,f}[j] \leq \mathbf{x}_{i,max}[j] \qquad \forall \mathbf{x}_i^{init} \in \mathcal{S}^{imp} \tag{5}$$

Constraint 1 asserts that the movement direction of the classifier is along dimension j. Constraint 2 asserts property i. Constraint 3 asserts property ii. Finally, constraints 4 and 5 assert property iii.

▶ **Theorem 23.** *Given the sets $\mathcal{S}^{yes}$, $\mathcal{S}^{no}$, and $\mathcal{S}^{imp}$, there is a polynomial-time algorithm that outputs a linear classifier (if one exists) that satisfies Properties i, ii,iii, or declares non-existence of such a classifier.*

Proof. If $\mathcal{S}^{\text{imp}} \neq \emptyset$, run LP 1-5 for all improvement dimensions j. If $\mathcal{S}^{\text{imp}} = \emptyset$, run the LP for $1 \leq j \leq n$. By Proposition 22, if there exist feasible solution $\mathbf{a}$ and b for one of these LPs, $f : \mathbf{ax} \geq b$ is a classifier satisfying properties i, ii, and iii. ◀

▶ **Corollary 24.** *There is a polynomial-time algorithm that determines whether there exists a linear classifier that classifies the initially qualified as positive, unqualified unimprovable agents as negative, encourages the agents in the improvement margin to improve to become qualified, and classifies them as positive. If such a classifier exists, it maximizes true positives subject to no false positives.*

▶ Remark 25. Theorem 5 asserts that given the initial feature vectors of agents, $\mathbf{x}_1^{\text{init}}, \mathbf{x}_2^{\text{init}}, \ldots, \mathbf{x}_n^{\text{init}} \in \mathbb{R}^d$, deciding whether there exists a classifier for which all the agents become true positives is NP-hard. However, when limiting to linear classifiers this problem is no longer NP-Hard. Using Theorem 23, by setting $\mathcal{S}^{\text{yes}}$ to the set of initially qualified agents, and $\mathcal{S}^{\text{imp}}$ to the rest of the agents, this problem is solvable in polynomial time.

5.3 Optimal General Classifier in Two-Dimensional Space

In this subsection, we consider the problem of maximizing true positives subject to no false positives in a 2-dimensional space, where the horizontal dimension is improvement, and the vertical dimension is gaming. We provide an algorithm in the linear model that given a set of agents, returns a set of target points $\mathcal{P}^{\text{final}} \subset \mathbb{R}^2$ that maximizes true positives subject to no false positives. Note that unlike Algorithm 1, our algorithm in this subsection does not take a finite set of target points $\mathcal{P}$ as input. For simplicity, by scaling we may assume wlog that $c = \mathbf{c}[1] = \mathbf{c}[2]$.

Overview of Algorithm 3. First, all the points $\mathbf{x}_i^{\text{init}}$ for $1 \leq i \leq m$ are sorted along the gaming dimension in a descending order, such that $\mathbf{x}_n^{\text{init}}$ has the smallest value in the gaming dimension. Our goal is to find designated points, $\mathbf{x'}_i$, for each $\mathbf{x}_i^{\text{init}}$. Starting with $\mathbf{x}_n^{\text{init}}$, for each point $\mathbf{x}_i^{\text{init}}$, move $\mathbf{x}_i^{\text{init}}$ along the improvement dimension until it crosses the line $\mathbf{a}^*\mathbf{x} = b^*$ at $\mathbf{x}_{i,min}$ (See Figure 4). Let $\mathbf{x'}_i$, the designated point of $\mathbf{x}_i^{\text{init}}$, be initially $\mathbf{x'}_i = \mathbf{x}_{i,min}$. If given the current set of designated points for agents $n, n-1, \ldots, i$, another point $\mathbf{x}_j^{\text{init}}$ for $j > i$ maximizes utility by moving to $\mathbf{x'}_i$ and becomes false positive, push $\mathbf{x'}_i$ upward along the gaming dimension, until $\mathbf{x}_j^{\text{init}}$ no longer picks $\mathbf{x'}_i$. When pushing $\mathbf{x'}_i$ along the gaming dimension, let $\mathbf{x}_{i,max}$ denote the furthest point that $\mathbf{x}_i^{\text{init}}$ can afford to reach to it. If the final point $\mathbf{x'}_i$ is such that $\mathbf{x}_i^{\text{init}}$ cannot afford to move to it, i.e. $\mathbf{x'}_i[2] > \mathbf{x}_{i,max}[2]$, discard $\mathbf{x'}_i$. Otherwise, $\mathbf{x'}_i$ is added to $\mathcal{P}^{\text{final}}$.

Note that we assume that if a point $\mathbf{x}_j^{\text{init}}$ can improve to $\mathbf{x'}_j$ and game to $\mathbf{x'}_i$ with the same cost, it would pick the improvement option.

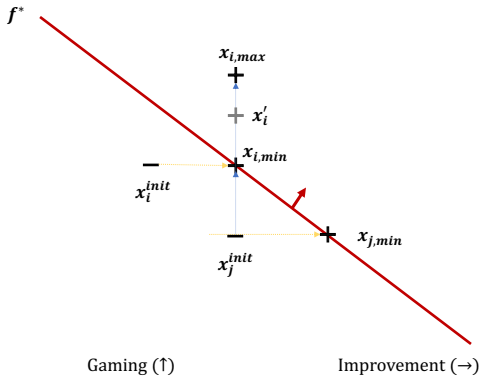

Figure 4 In Algorithm 3, $\mathbf{x'}_i$ is pushed along the gaming dimension so $\mathbf{x}_j^{\text{init}}$ no longer moves to it.

▶ **Theorem 26.** *Given initial feature vectors of agents, $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init} \in \mathbb{R}^2$, Algorithm 3 maximizes the number of true positives subject to no false positives.*

Proof. Proof is deferred to Appendix B. ◀

▶ Remark 27. By Corollary 6, this problem is NP-hard when $\mathcal{X} \subset \mathbb{R}^d$ for general (not constant) d.

Algorithm 3 Maximizing the number of true positives in 2-dimensions.

Input : $\mathcal{X}$, $f^* : \mathbf{a}^*\mathbf{x} \geq b^*$
Output : $\mathcal{P}^{\text{final}}$

Sort $\mathbf{x}_i \in \mathcal{X}$ in a descending order of $\mathbf{x}_i[2]$;
for $i = n, \cdots, 1$ **do**
  /* Let $\mathbf{x}_{i,min}$ be the projection of $\mathbf{x}_i$ on $\mathbf{a}^*\mathbf{x} = b^*$ along the improvement dimension */
  $\mathbf{x}_{i,min} = \left(\frac{b^* - \mathbf{a}^*[2]\mathbf{x}_i[2]}{\mathbf{a}^*[1]}, \mathbf{x}_i[2]\right)$;
  if $\mathbf{x}_{i,min}[1] - \mathbf{x}_i[1] > 1/c$ **then**
    /* $\mathbf{x}_i$ cannot become true positive. */
    continue;
  $\mathbf{x'}_i \leftarrow \mathbf{x}_{i,min}$;
  for $j = n, \cdots, i+1$ **do**
    if $cost(\mathbf{x}_j, \mathbf{x}'_j) > cost(\mathbf{x}_j, \mathbf{x}'_i)$ **then**
      $\mathbf{x}'_i \leftarrow (\mathbf{x}'_i[1], \mathbf{x}'_i[2] + cost(\mathbf{x}_j, \mathbf{x}'_j) - cost(\mathbf{x}_j, \mathbf{x}'_i))$;
  if $\mathbf{x'}_i[2] > \mathbf{x}_{i,max}[2]$ **then**
    /* $\mathbf{x}_i$ cannot become true positive without another point becoming false positive. */
    $\mathbf{x'}_i = (\mathbf{x'}_i[1], \infty)$;
  $\mathcal{P}^{\text{final}} \leftarrow \mathcal{P}^{\text{final}} \cup \mathbf{x'}_i$;
  return $\mathcal{P}^{\text{final}}$;

References

1 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 6–25, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465456.3467629.

2 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. On classification of strategic agents who can both game and improve. *arXiv preprint*, 2022. arXiv:2203.00124.

3 Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1774–1781, April 2020. doi:10.1609/aaai.v34i02.5543.

4 Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Causal feature discovery through strategic modification. *ArXiv*, abs/2002.07024, 2020. arXiv:2002.07024.

5 Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In *Proceedings of the 1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156 of *LIPIcs*, pages 9:1–9:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.FORC.2020.9.

6 Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 547–555, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2020408.2020495.

7 Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 55–70, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3219166.3219193.

8 Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

9 Alex M. Frankel and Navin Kartik. Improving information from manipulable data. *arXiv: Theoretical Economics*, June 2019. doi:10.1093/jeea/jvab017.

10 Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166. International Joint Conferences on Artificial Intelligence Organization, July 2020. Main track. doi:10.24963/ijcai.2020/23.

11 Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 111–122, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2840728.2840730.

12 Keegan Harris, Hoda Heidari, and Zhiwei Steven Wu. Stateful strategic regression. *CoRR*, abs/2106.03827, 2021. arXiv:2106.03827.

13 Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 259–268, New York, NY, USA, 2019. ACM. doi:10.1145/3287560.3287597.

14 Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pages 825–844, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3328526.3329584.

15 John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6917–6926. PMLR, 2020. URL: http://proceedings.mlr.press/v119/miller20b.html.

16 Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 230–239, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3287560.3287576.

17 Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Learning from strategic agents: Accuracy, improvement, and causality. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume abs/2002.10066 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 13–18 July 2020. URL: http://proceedings.mlr.press/v119/shavit20a.html, arXiv:2002.10066.

18 Shenke Xiao, Zihe Wang, Mengjing Chen, Pingzhong Tang, and Xiwang Yang. Optimal common contract with heterogeneous agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7309–7316, April 2020. doi:10.1609/aaai.v34i05.6224.

A Missing Proofs of Section 3

Proof of Proposition 1. The size of $\mathcal{X}$ is n, and within the for loop each computation takes $O(1)$ time since the edges for each x_i are already sorted. When the flag is set to 1, at least one point in $\mathcal{P}$ is removed, and when the flag is 0 at the end of the inner loop, the algorithm returns. Therefore, the outer loop is run at most $|\mathcal{P}|$ times while the inner loop is run n times; resulting in a running time of $O(|\mathcal{P}|n)$. ◀

Proof of Theorem 2. Let A be the improving agents (agents taking blue edges) associated with the set of criteria $\mathcal{P}^{\text{final}}$. We show that having any other set $Q \subseteq \mathcal{P}$ as the criteria, either causes an agent to take a red edge, or no more than $|A|$ agents to take blue edges. To do so, consider partitioning Q into two subsets Q^F and $Q^{\bar{F}}$, where $Q^F \subseteq \mathcal{P}^{\text{final}}$ and $Q^{\bar{F}} \subseteq \mathcal{P} \setminus \mathcal{P}^{\text{final}}$.

First, we show that if $Q^{\bar{F}} \neq \emptyset$, an agent takes a red edge. To prove this claim, suppose by contradiction that $Q^{\bar{F}}$ is nonempty and consider the first time the algorithm deletes an element $p \in Q^{\bar{F}}$. At this stage, the set of criteria in the algorithm $\mathcal{P}'$ is a superset of

$Q^{\bar{F}} \cup \mathcal{P}^{\text{final}}$. By definition, p is the lowest-weight neighbor of a gaming agent, a, in $\mathcal{P}'$. This implies that p is also the lowest-weight neighbor of a in $Q \subseteq Q^{\bar{F}} \cup \mathcal{P}^{\text{final}} \subseteq \mathcal{P}'$, and a is a gaming agent given the criteria set Q. This implies the claim.

Secondly, we show that among the sets of criteria with no gaming agent, $\mathcal{P}^{\text{final}}$ has the highest number of improving agents. The previous claim implies that any set of criteria with no gaming agent is a subset of $\mathcal{P}^{\text{final}}$. Now, we need to show that among $Q \subseteq \mathcal{P}^{\text{final}}$, $\mathcal{P}^{\text{final}}$ has the largest set of improving agents. This is trivial, since by considering a subset we may only lose on agents in A that do not have a neighbor in Q or their lowest-weight edge is red. Therefore, any $Q \subseteq \mathcal{P}^{\text{final}}$ has at most $|A|$ improving agents. ◀

Proof of Theorem 4. We show the following problem is NP-hard.

▶ **Problem 2.** *Suppose we are given a set of n agents where $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init}$ denote their initial feature vectors, and a set $\mathcal{P}$ of potential criteria also called target points in the linear model. Find a subset $\mathcal{P}^{final} \subseteq \mathcal{P}$ that maximizes the number of true positives subject to at most k false positives.*

We prove the NP-hardness by reducing the Max-k-Cover problem with equal-sized sets of size 3 to this problem. In the Max-k-Cover problem, we are given a set $\mathcal{E}$ of elements e_i, and sets $S_j \subseteq \mathcal{E}$, and the goal is to select at most k sets out of S_j that maximize the number of elements they cover.

First, we show how to construct an instance of Problem 2 from an instance of the Max-k-Cover problem. To do so, we determine the number of dimensions, initial positions of the agents, the target points, and the movement costs. Let n be the number of elements of the Max-k-Cover instance, we construct an $n+1$-dimensional space where the first n dimensions are improvement and the last dimension is gaming. Consider elements $e_1, e_2, \ldots, e_n$ in the Max-k-Cover instance. For every element, we consider an agent; and for every set, we consider an agent and a target point. For e_i, the corresponding agent is at initial point $\mathbf{x}_i^{\text{init}}$, an $n+1$-dimensional vector whose i^{th} and $n+1^{st}$ coordinates are 1 and the other coordinates are 0. For every set S_j, we consider a target point $\mathbf{p}_j$ and an agent with initial point $\mathbf{x}_{n+j}^{\text{init}}$. In $\mathbf{p}_j$, the coordinates corresponding to the elements in S_j and the $n+1^{st}$ coordinate are set to 1 and the rest of the coordinates are 0. In $\mathbf{x}_{n+j}^{\text{init}}$, the coordinates corresponding to the elements in S_j are set to 1, the $n+1^{st}$ coordinate is set to -1, and the rest of the coordinates are 0. Finally, let the movement cost in any dimension be $1/2$. Note that this construction fits into the framework of a linear model and $f^* : \sum_{j=1}^{n+1} \mathbf{x}[j] \geq 4$ is the linear threshold function for the truly qualified agents. All the target points $\mathbf{p}_j$ satisfy the threshold and all the agents are initially unqualified and do not meet the threshold.

Next, we discuss what target point each agent selects and whether they become truly qualified (true positive) or not (false positive). Because the cost per unit of movement equals $1/2$, each agent can only afford to reach to target points with distance at most 2. Agents $\mathbf{x}_i^{\text{init}}$ for $i \in \{1, \ldots, n\}$ can only afford to reach a target point whose i^{th} coordinate is 1 since they are at distance 2. They are at distance 3 to any other target points. Since all dimensions $1, \ldots, n$ are improving dimensions these agents become truly qualified when they reach such target points. Agents $\mathbf{x}_i^{\text{init}}$ for $i > n$ can only afford to reach $\mathbf{p}_i$ since they have distance 2. They have distance more than 2 to any other target points. Agents $\mathbf{x}_i^{\text{init}}$ for $i > n$ can only reach to $\mathbf{p}_i$. To do so, these agents move in a gaming dimension and do not become truly qualified.

Finally, we show how the solutions of these two problems coincide. Consider the problem of maximizing the true positives subject to including at most k false positives. Including each $\mathbf{p}_j$ in the final set of target points, $\mathcal{P}^{\text{final}}$, causes exactly one agent, $\mathbf{x}_j^{\text{init}}$, to be a false

positive. Therefore, having at most k false positive is equivalent to including at most k target points. Maximizing the true positives subject to at most k target points is exactly equivalent to selecting at most k sets that maximize the elements they cover. This completes the reduction. ◀

Proof of Theorem 5. We show the following problem is NP-hard.

▶ **Problem 3.** *Suppose we are given a set of n agents where $\mathbf{x}_1^{init}, \mathbf{x}_2^{init}, \ldots, \mathbf{x}_n^{init}$ denote their initial feature vectors. Does there exist a set of target points $\mathcal{P}^{final} \subseteq \mathbb{R}^d$ for which all the agents become truly qualified?*

We prove the NP-hardness by a reduction from the approximate version of the hitting set with equal-sized sets problem. As an instance of the hitting set problem we are given, $(\mathcal{F}, \mathcal{E})$ where $\mathcal{F} = \{S_1, \cdots, S_m\}$ is a collection of the subsets of $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$, and each set S_i has a size of $0 < s < n$, and our goal is to find a minimum size set $S^* \subseteq \mathcal{E}$ that intersects every set in $\mathcal{F}$. In order to show NP-hardness, we construct an instance of Problem 3 and prove: (1) If all the agents can become true positives by reaching to a set of target points $\mathcal{P}^{\text{final}} \subseteq \mathbb{R}^d$ that the mechanism designer selects, then there exists a hitting set of size at most $2k$. (2) If there exists a hitting set of size k then the mechanism designer can select a set of target points that encourages all the agents to become true positives. Since hitting set and set cover problems are equivalent and approximating set cover within a constant factor is NP-hard [8], this implies that Problem 3 is NP-hard.

First, we show how to construct an instance of Problem 3 from an instance of the Hitting Set problem. To do so, we determine the number of dimensions, initial positions of the agents, the movement costs, and a linear threshold function for the truly qualified. Let n be the number of elements of the Hitting Set instance, we construct an $n+1$-dimensional space where the first n dimensions are improvement and the last dimension is gaming. Consider sets $S_1, S_2, \ldots, S_m$ in the Hitting Set instance. For every set S_i, we consider agent i at initial point $\mathbf{x}_i^{\text{init}}$. In $\mathbf{x}_i^{\text{init}}$, the j^{th} coordinates such that $e_j \in S_i$ is set to 1. The rest of the first n coordinates are set to $2k$ and the last coordinate is 0. Also consider an extra agent $m+1$ at initial point $\mathbf{x}_{m+1}^{\text{init}}$ where all the first n coordinates are 0 and the last coordinate is $2k(n-s)+s$. Note that for all the agents $\sum_{j=1}^{n+1} \mathbf{x}_i^{\text{init}}[j] = 2k(n-s)+s$. Let the movement cost in all the dimensions $1 \le j \le n$ be $\frac{1}{2k}$ and in dimension $n+1$ be c such that $\frac{1}{2k(n-s)+s+1} < c < \frac{1}{2k(n-s)+s}$. Let $f^* : \sum_{j=1}^{n+1} \mathbf{x}[j] \ge 2k(n-s)+s+2k$. Therefore, all the agents are initially unqualified and at ℓ_1 distance of $2k$ from f^*.

Now we prove the first direction, i.e., if all the agents can become true positives by reaching to a set of target points $\mathcal{P}^{\text{final}} \subseteq \mathbb{R}^d$ that the mechanism designer selects, then there exists a hitting set of size at most $2k$. For all $1 \le i \le m+1$, let $\mathbf{p}_i \in \mathcal{P}^{\text{final}}$ denote the target point that $\mathbf{x}_i^{\text{init}}$ moves to and becomes true positive.

It consists of the following arguments: (i) For all $1 \le i \le m+1$, agent i receives utility 0 by reaching to $\mathbf{p}_i$. (ii) For all $1 \le i \le m$, agent $m+1$ does not afford to reach to $\mathbf{p}_i$. (iii) If $\mathbf{p}_{m+1}[j] \le 1$ for all $e_j \in S_i$, agent i moves to $\mathbf{p}_{m+1}$ and becomes a false positive. Therefore if all agents improve, for each $1 \le i \le m$, there exists $e_j \in S_i$ such that $\mathbf{p}_{m+1}[j] > 1$. (iv) In order for agent $m+1$ to afford to reach to target point $\mathbf{p}_{m+1}$, the number of coordinates $1 \le j \le m$ with value at least 1 must be at most $2k$. (v) These elements constitute a hitting set of size at most $2k$.

First, we prove argument (i). Each agent $1 \le i \le m+1$, is at ℓ_1 distance of $2k$ to f^*. To become qualified it needs to move $2k$ in the improvement dimensions. Since moving for a distance of $2k$ along the improvement dimensions costs a value of $(2k) \times (\frac{1}{2k}) = 1$, agent i makes a utility of 0.

Now, we move to argument (ii). Following up on the previous claim, to reach $\mathbf{p}_i$, agent $1 \leq i \leq m$ spends all of their movement budget in the improvement dimensions and cannot move a positive amount in the gaming dimension $n+1$. Therefore, $\mathbf{p}_i[n+1] = 0$ and $\sum_{j=1}^{n} \mathbf{p}_i[j] = 2k(n-s) + s + 2k$. In order for agent $m+1$ to reach such a target point, it needs to move a total of $2k(n-s) + s + 2k > 2k$ in the improvement dimensions, which costs more than 1 and it cannot afford.

Next, we prove argument (iii). Since $\mathbf{x}_{m+1}^{\text{init}}$ has an ℓ_1 distance of $2k$ from f^* and costs exactly a value of 1 to reach there, it can only afford to move along the improvement dimensions. Therefore, $\mathbf{p}_{m+1}[n+1] \leq 2k(n-s)+s$. Additionally, for $1 \leq j \leq n$, $\mathbf{p}_{m+1}[j] \leq 2k$; otherwise, agent $m+1$ cannot afford to reach to $\mathbf{p}_{m+1}$. Suppose $\mathbf{p}_{m+1}[j] \leq 1$ for all $e_j \in S_i$. Using this assumption, for agent i to reach $\mathbf{p}_{m+1}$ it only needs to pay cost of movement in dimension $n+1$, moving $2k(n-s)+s$ units and paying c per unit of movement. Since $(2k(n-s)+s) \times c < 1$, agent i makes a strictly positive utility. Therefore agent i prefers $\mathbf{p}_{m+1}$ over any other target point that makes it true positive which by argument (i) achieves utility 0.

Argument (iv) is straight-forward. To achieve non-negative utility each agent can afford to move at most $2k$ units along the improvement dimensions. Therefore, for the target point $\mathbf{p}_{m+1}$, the number of coordinates $1 \leq j \leq n$ with value at least 1 must be at most $2k$.

Argument (v) is a direct implication of the two previous arguments. By argument (iii), for each $1 \leq i \leq m$ there is an element $e_j \in S_i$ such that $\mathbf{p}_{m+1}[j] > 1$. By argument (iv), the number of coordinates $j \leq n$ such that $\mathbf{p}_{m+1}[j] > 1$ is at most $2k$ since otherwise agent $m+1$ cannot afford to reach to $\mathbf{p}_{m+1}$. Therefore, elements e_j such that $\mathbf{p}_{m+1}[j] > 1$ constitute a hitting set of size at most $2k$.

Now, we prove the reverse direction: if there exists a hitting set S^* of size k, the mechanism designer can select a set of target points that encourages all the agents to become true positives. To do so, we construct a set of target points $\mathcal{P}^{\text{final}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{m+1}\}$ that makes every agent to become true positive. For each agent i, $1 \leq i \leq m$, put a target point $\mathbf{p}_i$ whose first coordinate is $2k$ more than $\mathbf{x}_i^{\text{init}}$. For agent $m+1$, put a target point $\mathbf{p}_{m+1}$ whose coordinates j where $e_j \in S^*$ are set to 2 and the remaining agree with $\mathbf{x}_{m+1}^{\text{init}}$. Each target point $\mathbf{x}_i^{\text{init}}$ is set such that $\sum_{j=1}^{n+1} \mathbf{x}_i^{\text{init}}[j] = 2k(n-s) + s + 2k$. In order to show that every agent is able to improve, we argue that: (i) For all $1 \leq i \leq m$, agent i can afford to move to $\mathbf{p}_i$. Additionally, if agent i moves to any of the target points $\mathbf{p}_j$ where $1 \leq j \leq m$, it becomes true positive. (ii) For all $1 \leq i \leq m$, agent i cannot reach to $\mathbf{p}_{m+1}$. (iii) Agent $m+1$ moves to $\mathbf{p}_{m+1}$ and becomes true positive.

First, we prove argument (i): Agent i is at a distance of $2k$ from $\mathbf{p}_i$. It can afford to reach to $\mathbf{p}_i$ by paying a cost of $(2k) \times (\frac{1}{2k}) = 1$ and become true positive. In addition, if it moves to any of the other target points $\mathbf{p}_j$ where $1 \leq j \leq m$, since it has only moved along the improvement dimensions, it would become true positive.

Next, we prove argument (ii): We know that for each S_i, there exists an element $e_j \in S_i$ such that $\mathbf{p}_{m+1}[j] = 2$. As a result, the ℓ_1 distance of $\mathbf{x}_i^{\text{init}}$ and $\mathbf{p}_{m+1}$ is at least $(2k(n-s) + s + 1) \times c > 1$. Therefore, for each $1 \leq i \leq m$, $\mathbf{x}_i^{\text{init}}$ cannot afford to reach to $\mathbf{p}_{m+1}$.

Finally, we prove argument (iii): First, we argue that agent $m+1$ cannot afford to reach to any of the target points $\mathbf{p}_i$ where $1 \leq i \leq m$. For each target point $\mathbf{p}_i$ where $1 \leq i \leq m$, $\mathbf{p}_i[n+1] = 0$ and $\sum_{j=1}^{n} \mathbf{p}_i[j] = 2k(n-s) + s + 2k$. In order for agent $m+1$ to reach such a target point, it needs to move a total of $2k(n-s) + s + 2k > 2k$ units in the improvement dimensions, which costs more than 1 and it cannot afford. In addition, agent $m+1$ can afford to move to $\mathbf{p}_{m+1}$, and by reaching there it becomes true positive.

As a result of the above arguments, given a hitting set of size k, the mechanism designer can select a set of target points that encourages all the agents to become true positives.

Combining the above two directions, shows that the problem of selecting a set of target points for which all the agents become truly qualified is NP-hard. ◀

B Proof of Theorem 26

In order to prove Theorem 26, we need to first show that the following observation and lemma hold.

▶ **Observation 28.** *Line $\mathbf{a}^*\mathbf{x} = b^*$ has a negative slope, i.e., each feature is defined so that larger is better. Therefore, after the points in $\mathcal{X}$ are sorted, if an agent $\mathbf{x}_j^{init}$ where $j < i$ reaches to any point $\mathbf{x'}_i \in [\mathbf{x}_{i,min}, \mathbf{x}_{i,max}]$, then $\mathbf{x}_j^{init}$ becomes true positive. On the other hand, for $j > i$, if $\mathbf{x}_j^{init}$ moves to any point $\mathbf{x'}_i \in [\mathbf{x}_{i,min}, \mathbf{x}_{i,max}]$, then $\mathbf{x}_j$ becomes false positive.*

▶ **Lemma 29.** *Consider a point $\mathbf{p}$ such that $\mathbf{p}[1] \geq \mathbf{x}_{i,min}[1]$, and another point $\mathbf{q} \in [\mathbf{x}_{i,min}, \mathbf{x}_{i,max}]$. Suppose $cost(\mathbf{x}_i^{init}, \mathbf{p}) = cost(\mathbf{x}_i^{init}, \mathbf{q})$. Then, for any $j > i$, it is the case that $cost(\mathbf{x}_j^{init}, \mathbf{p}) \leq cost(\mathbf{x}_j^{init}, \mathbf{q})$.*

Proof. Initially, if $\mathbf{p}[2] < \mathbf{x}_i^{\text{init}}[2]$, $\mathbf{p}$ is replaced with $(\mathbf{p}[1], \mathbf{x}_i^{\text{init}}[2])$. By doing so, $cost(\mathbf{x}_j^{\text{init}}, \mathbf{p})$ would not decrease. Hence, without loss of generality, we can assume $\mathbf{p}[2] \geq \mathbf{x}_i^{\text{init}}[2]$.

First, we show that $cost(\mathbf{x}_j^{\text{init}}, \mathbf{p}) \leq cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{p})$, where the inequality holds when $\mathbf{x}_{i,min}[1] < \mathbf{x}_j^{\text{init}}[1] \leq \mathbf{p}[1]$.

$$
\begin{aligned}
&cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{p}) \\
&= max\Big\{\mathbf{x}_{i,min}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + \Big(\mathbf{x}_{i,min}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) + \Big(\mathbf{p}[1] - \mathbf{x}_{i,min}[1]\Big) + \Big(\mathbf{p}[2] - \mathbf{x}_{i,min}[2]\Big) \\
&= max\Big\{\mathbf{x}_{i,min}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + \Big(\mathbf{p}[1] - \mathbf{x}_{i,min}[1]\Big) + \Big(\mathbf{p}[2] - \mathbf{x}_j^{\text{init}}[2]\Big)
\end{aligned}
$$

If $\mathbf{x}_j^{\text{init}}[1] \leq \mathbf{x}_{i,min}[1]$, the last equation above gets equal to $\Big(\mathbf{p}[1] - \mathbf{x}_j^{\text{init}}[1]\Big) + \Big(\mathbf{p}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) = cost(\mathbf{x}_j^{\text{init}}, \mathbf{p})$. Otherwise, $\mathbf{x}_j^{\text{init}}[1] > \mathbf{x}_{i,min}[1]$ and the last equation above gets equal to $\Big(\mathbf{p}[1] - \mathbf{x}_{i,min}[1]\Big) + \Big(\mathbf{p}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) > \Big(\mathbf{p}[1] - \mathbf{x}_j^{\text{init}}[1]\Big) + \Big(\mathbf{p}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) = cost(\mathbf{x}_j^{\text{init}}, \mathbf{p})$. In any case, $cost(\mathbf{x}_j^{\text{init}}, \mathbf{p}) \leq cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{p})$.

Next we argue that $cost(\mathbf{x}_{i,min}, \mathbf{p}) = cost(\mathbf{x}_{i,min}, \mathbf{q})$. First, since $\mathbf{p}[1] \geq \mathbf{x}_{i,min}[1]$ and $\mathbf{p}[2] \geq \mathbf{x}_{i,min}[2]$, then $cost(\mathbf{x}_i, \mathbf{p}) = cost(\mathbf{x}_i, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{p})$. Similarly, $cost(\mathbf{x}_i, \mathbf{q}) = cost(\mathbf{x}_i, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{q})$. Since $cost(\mathbf{x}_i, \mathbf{p}) = cost(\mathbf{x}_i, \mathbf{q})$, it is the case that $cost(\mathbf{x}_{i,min}, \mathbf{p}) = cost(\mathbf{x}_{i,min}, \mathbf{q})$.

Therefore,

$$
\begin{aligned}
&cost(\mathbf{x}_j^{\text{init}}, \mathbf{p}) \leq cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{p}) \\
&\leq cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_{i,min}) + cost(\mathbf{x}_{i,min}, \mathbf{q}) \\
&= max\Big\{\mathbf{x}_{i,min}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + max\Big\{\mathbf{x}_{i,min}[2] - \mathbf{x}_j^{\text{init}}[2], 0\Big\} + \\
&max\Big\{\mathbf{q}[1] - \mathbf{x}_{i,min}[1], 0\Big\} + max\Big\{\mathbf{q}[2] - \mathbf{x}_{i,min}[2], 0\Big\} \\
&= max\Big\{\mathbf{x}_{i,min}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + \Big(\mathbf{x}_{i,min}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) + \Big(\mathbf{q}[2] - \mathbf{x}_{i,min}[2]\Big) \\
&= max\Big\{\mathbf{x}_{i,min}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + \Big(\mathbf{q}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) \\
&= max\Big\{\mathbf{q}[1] - \mathbf{x}_j^{\text{init}}[1], 0\Big\} + \Big(\mathbf{q}[2] - \mathbf{x}_j^{\text{init}}[2]\Big) \\
&= cost(\mathbf{x}_j^{\text{init}}, \mathbf{q})
\end{aligned}
$$

◀

Proof of Theorem 26. Suppose not. Let $\mathbf{x}_1^{\text{OPT}}, \ldots, \mathbf{x}_n^{\text{OPT}}$ be an optimal solution that agrees with $\mathbf{x'}_1, \ldots, \mathbf{x'}_n$ on as large a suffix as possible, and let i be the largest index such that $\mathbf{x}_i^{\text{OPT}} \neq \mathbf{x}'_i$ (so $\mathbf{x}_j^{\text{OPT}} = \mathbf{x}'_j$ for all $j > i$).

First, note that $i \neq n$. This is because $\mathbf{x'}_n = \mathbf{x}_{n,min}$, which is the cheapest point that agent n can reach to become a true positive; moreover, any other point moving to $\mathbf{x}'_n$ is a true improvement. So, replacing $\mathbf{x}_n^{\text{OPT}}$ with $\mathbf{x'}_n$ only helps.

Next, we claim that even if $i < n$, replacing $\mathbf{x}_i^{\text{OPT}}$ with $\mathbf{x'}_i$ can only improve the optimal solution. First, if $cost(\mathbf{x}_i^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) \geq cost(\mathbf{x}_i^{\text{init}}, \mathbf{x'}_i)$ then replacing $\mathbf{x}_i^{\text{OPT}}$ with $\mathbf{x'}_i$ only helps by the same argument as above and the fact that $\mathbf{x'}_i$ was chosen so that no agent $j > i$ manipulates to it; here we are using the fact that the suffixes of the two solutions agree. On the other hand, suppose that $cost(\mathbf{x}_i^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) < cost(\mathbf{x}_i^{\text{init}}, \mathbf{x'}_i)$ and $cost(\mathbf{x}_i^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) \leq 1/c$. Since $\mathbf{x}_i^{\text{init}}$ cannot become a false positive by moving to $\mathbf{x}_i^{\text{OPT}}$, this means that $\mathbf{x}_i^{\text{OPT}}[1] \geq \mathbf{x}_{i,min}[1]$. There exists a point $\mathbf{q} \in [\mathbf{x}_{i,min}, \mathbf{x}_{i,max}]$ such that $cost(\mathbf{x}_i^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) = cost(\mathbf{x}_i^{\text{init}}, \mathbf{q})$, which implies that $cost(\mathbf{x}_i^{\text{init}}, \mathbf{q}) < cost(\mathbf{x}_i^{\text{init}}, \mathbf{x'}_i)$. The reason that $\mathbf{q}$ was not selected as $\mathbf{x'}_i$ is that there exists an agent $\mathbf{x}_j^{\text{init}}$ where $\mathbf{x}_j^{\text{init}}$ moves to $\mathbf{q}$ and becomes false positive. By Observation 28, $j > i$. Hence, $cost(\mathbf{x}_j^{\text{init}}, \mathbf{q}) < cost(\mathbf{x}_j^{\text{init}}, \mathbf{x'}_j)$ and $cost(\mathbf{x}_j^{\text{init}}, \mathbf{q}) \leq 1/c$. By Lemma 29, $cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) \leq cost(\mathbf{x}_j^{\text{init}}, \mathbf{q})$, so $cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) < cost(\mathbf{x}_j^{\text{init}}, \mathbf{x'}_j)$ and $cost(\mathbf{x}_j^{\text{init}}, \mathbf{x}_i^{\text{OPT}}) \leq 1/c$. Hence, $\mathbf{x}_j^{\text{init}}$ is closer to $\mathbf{x}_i^{\text{OPT}}$ compared to $\mathbf{x'}_j = \mathbf{x}_j^{\text{OPT}}$ and so agent j would become a false positive under OPT, which contradicts the definition of OPT. So, this second case cannot occur.

Therefore, Algorithm 3 maximizes the number of true positives subject to having no false positives. ◀

Individually-Fair Auctions for Multi-Slot Sponsored Search

Shuchi Chawla
Department of Computer Science, University of Texas at Austin, TX, USA

Rojin Rezvan
Department of Computer Science, University of Texas at Austin, TX, USA

Nathaniel Sauerberg
Department of Computer Science, University of Texas at Austin, TX, USA

Abstract

We design fair sponsored search auctions that achieve a near-optimal tradeoff between fairness and quality. Our work builds upon the model and auction design of Chawla and Jagadeesan [5], who considered the special case of a single slot. We consider sponsored search settings with multiple slots and the standard model of click through rates that are multiplicatively separable into an advertiser-specific component and a slot-specific component. When similar users have similar advertiser-specific click through rates, our auctions achieve the same near-optimal tradeoff between fairness and quality as in [5]. When similar users can have different advertiser-specific preferences, we show that a preference-based fairness guarantee holds. Finally, we provide a computationally efficient algorithm for computing payments for our auctions as well as those in previous work, resolving another open direction from [5].

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design

Keywords and phrases algorithmic fairness, advertising auctions, and individual fairness

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.4

Related Version *Full Version*: `https://arxiv.org/abs/2204.04136`

Funding This work was supported in part by NSF Award CCF-2008006.

1 Introduction

We study the design of ad auctions under a fairness constraint. Fairness in the context of sponsored content has received considerable attention in recent years. It has been observed, for example, that ads on platforms such as Facebook and Google disproportionately target certain demographics, discriminating across users on the basis of race and gender. Furthermore, standard auction formats such as highest-bids-win can lead to discrimination even when the input to these algorithms, namely bids, CTRs, and relevance scores are themselves non-discriminatory.

[4] initiated the study of optimal auction design under the constraint that the auction does not add any unfairness beyond what is already present in bids, and proposed a class of *proportional allocation* algorithms as a solution that achieves fairness while also providing an approximation to the optimal social welfare. In a followup work, [5] designed a class of *inverse proportional allocation* algorithms and showed that this class of mechanisms achieves an optimal tradeoff between social welfare and fairness. Both of these works focused on the simple case of a single item auction and left open the problem of designing a fair and efficient multi-slot position auction.

In this paper we extend the design of fair auctions from the single item setting to arbitrary position auction settings. We show that both the proportional allocation and inverse proportional allocation algorithms can be adapted to the setting of a position auction

while inheriting their single-unit fairness properties as well as their approximation to social welfare. As in [4, 5] our auctions provide fair solutions when the advertisers' bids are themselves non-discriminatory. Auctions for multi-slot settings must take into account both the advertisers' preferences over users as captured by per-click values, as well as the users' preferences over advertisers as captured by click through rates. We consider two different models for formalizing fairness in these settings. In the first, we consider differences of allocation across users that are close both in terms of the values advertisers assign to them as well as in terms of their own click through rates; we require that such users receive similar allocations. In the second setting, we consider pairs of users that are similarly qualified as per advertisers' values, but have different preferences (i.e. CTRs). In this case, while the users may receive different allocations, we require that allocations are suitably aligned with users' preferences. We elaborate on the details of these models below. Finally, we address another open question in [4, 5] and show how to efficiently compute supporting prices for both proportional and inverse proportional allocation.

Formalizing fairness across users

Consider two users Alice and Bob who are similar in most respects but differ in a sensitive demographic such as gender or race. Individual fairness then posits that Alice and Bob should see similar ad allocations. For example, it would be unfair to show more employment ads to Bob and more online retail ads to Alice. One potential source of unfairness in ad allocations is the use of discriminatory targeting by advertisers. However, empirical studies as well as theoretical analysis shows that unfairness in allocations can persist even in the absence of discriminatory targeting. The culprit is allocation algorithms that turn minor differences in advertisers' bids into large swings in allocation. Suppose, for example, that an employment agency places a slightly higher value on Bob than on Alice whereas an online retail store places a slightly higher value on Alice because of minor differences in the users' profiles. Then the highest-bid-wins auction would show entirely different ads to the two users.

To combat this problem, [5] formalize the notion of fairness in auctions as a "value stability" constraint. Informally speaking, value stability requires that whenever two users receive multiplicatively similar values from all advertisers (such as Alice and Bob in the example above) they must receive close allocations (as measured in terms of the ℓ_∞ distance between the respective probability distributions over the ad displayed). Previous work shows that while optimal auctions do not satisfy value stability, there are simple auction formats that do. In the *Proportional Allocation* (PA) mechanism, allocations are proportional to (some increasing function of) the advertisers' reported values. In the *Inverse Proportional Allocation* (IPA) mechanism, the unallocated amounts, i.e., one minus the probability of allocation, are inversely proportional to (some increasing function of) the advertisers' reported values. In both mechanisms, the allocation is a sufficiently smooth function of the advertisers' values and therefore satisfies some form of value stability. We mostly focus on the IPA mechanism in this paper as it provides better tradeoffs between fairness and welfare.

Multi-slot extensions

As a simple extension of the single slot setting, consider a setting with k slots, where each ad and each slot are equally likely to be clicked by the user, so the relative placement of ads in slots does not matter. In this case, one straightforward way to to extend the single-slot allocations is to simply multiply them by k; if this provides a valid allocation, the fairness and

welfare guarantees follow immediately from the single-slot case. The problem is that some ads may receive a total allocation greater than 1 and simply capping allocations at 1 breaks the fairness guarantee. We propose a different extension of the IPA. As in the single slot case, we ensure that the unallocated amounts to advertisers are inversely proportional to (some function of) the reported values, subject to the total allocation equaling k. The fairness a.k.a. value stability of this extension follows easily from the single-slot special case. We further show that the social welfare approximation of multi-slot IPA matches its approximation for the single-item case by characterizing worst case instances for the approximation factor.

While the above discussion provides a complete story for the case of a multi-unit auction, in the case of online advertising, we also need to take click through rates into account. Throughout this paper, we assume that click through rates are multiplicatively separable into ad-specific and slot-specific components. In other words, the click through rate of an ad i placed in slot j is given by $\alpha_i \times \beta_j$ for some parameters α and β specific to each user that are known to the platform/auctioneer. We further assume that all users weakly prefer earlier slots to later slots. Under these assumptions, we present an extension of the IPA to the ad auction setting that exactly maintains the social welfare guarantees of their single- and multi-unit counterparts. In particular, the social welfare approximation is independent of the number of slots.

Fairness in the context of click through rates

is tricky to define, however. As before, we may assume that if two users are similarly qualified for all ads but differ in their sensitive attributes, then the two users receive multiplicatively similar per-click values from all advertisers. However, click through rates capture the users' own preferences and similar users may not have similar click through rates. What sort of fairness guarantees can we then provide?

We first show that differences in slot-specific CTRs do not impact fairness guarantees.[1] In particular, two users with similar values and similar ad-specific CTRs α receive allocations that are close in ℓ_∞ distance. In particular, the probability of assigning any particular slot to any particular ad is additively close for the two users. In fact, this additive closeness holds also for the probability that any particular ad is assigned to slot j or better for any j.

We then consider settings with similarly qualified users that have arbitrarily different ad-specific and slot-specific CTRs. Observe that in order to achieve any reasonable guarantee for social welfare, our allocation algorithms must take ad-specific CTRs into account. As a result, it is impossible to provide a value-stability guarantee in this setting while also providing an approximation to social welfare. Nevertheless, we show that a form of preference-aligned fairness holds. Specifically, let Alice and Bob be two users with multiplicatively similar values and let α and α' denote their ad-specific CTR vectors. Then we show that although the two users' allocations can be quite far from each other, Alice receives a higher allocation than Bob for precisely the ads that she is more likely to click on, and vice versa. Formally, if we sort the advertisers in decreasing order of the ratio α_i/α'_i, then for every i, the probability that Alice gets to see an ad with index $\leq$ i is at least as large as Bob's probability of seeing the same set of ads.

[1] In fact, the allocations produced by our algorithms do not depend on the slot-specific CTRs, although the payments made by advertisers necessarily must.

Computing payments

We conclude our study with a discussion of payments. It is easy to observe that both generalized IPA and generalized PA have monotone allocation rules in the advertisers' reported values. However, computing the supporting prices is not straightforward and was left open in previous work. Let $x_i(v_i)$ denote the net allocation (expected probability of click) to advertiser i for a particular user, when the advertiser reports a per-click value of v_i. We show that $x_i(v_i)$ is a piecewise rational function with polynomially many pieces and that it is possible to compute the functional form of each piece in polynomial time. Computing payments using Myerson's lemma then boils down to computing polynomially many integrals over rational functions.

Organization of the paper

We present our extension of the IPA in Section 3 and prove its social welfare and fairness guarantees for the setting of similarly qualified users with similar preferences. In Section 4 we discuss fairness for users that are similarly qualified but have different preferences. Section 5 presents our algorithm for computing payments. We extend our results to the PA in Section 6. Most proofs are deferred to the appendix or removed due to space limitations. [2]

Related Work

Journalism and empirical work have revealed the myriad ways in which existing ad auction systems lead to unfairness and discrimination [2, 10, 11, 12, 14]. One approach to addressing these issues develops advertiser strategies for bidding in existing auction formats while ensuring statistical parity between groups [9, 15].

More related to our approach is theoretical work on designing auctions and, more generally, algorithms that guarantee fairness properties. These fairness properties typically differ in two dimensions: 1) whether they apply to individuals or only to groups as a whole, and 2) whether they enforce fairness by similarity of treatment or outcome, satisfaction of preferences (e.g., in the form of envy-freeness), or something bridging the two.

These notions of fairness grew out of the fair classification literature, where Dwork et al. [6] were the first to propose an individual fairness notion requiring agents who are similar under some task-specific metric to receive similar classifications. Dwork and Ilvento investigate in [7] whether compositions of such classification algorithms that are fair in isolation maintain their fairness properties.

Kim et al. [13] introduce individual preference-informed fairness by augmenting this notion of individual fairness with envy-freeness, allowing the allocations of similar users to differ in accordance with their preferences. Similarly, Zafar et al. in [18] develop notions of preference-informed group fairness by allowing deviations from parity in treatment and impact if the deviations are envy-free.

Our work employs and expands upon a model of individual fairness in sponsored search first developed by [4] and based on the multi-category fairness work of [7]. An alternate model, also based on [7], was presented by [16], albeit in a Bayesian setting. A main difference between our work and [16] is that we study the design of auctions that achieve an optimal tradeoff between fairness and welfare, whereas [16] analyzes the fairness and welfare of two specific mechanisms. Another relevant work is that of [8] who study the fairness-welfare

[2] For the full version, visit `https://arxiv.org/abs/2204.04136`.

tradeoff in a Bayesian setting. [8] draws a connection between individual fairness in this context and multi-item auctions with an item symmetry constraint, giving simple mechanisms that achieve a constant-approximation to the revenue-optimal fair mechanism.

There is also some recent work on group-fair ad auctions, such as [17], which shows that constraints on advertiser behavior which enforce group fairness notations can actually increase the profit of the platform. In a Bayesian setting, [3] augments generalized second price auctions with fair division schemes to achieve good social welfare guarantees while satisfying envy-freeness properties among advertiser groups.

As far as we know, ours is the first work addressing fairness specifically in the positional auctions setting where different users have different click through rates.

2 Models and Definitions

We consider the following stylized model for online advertising auctions. Let U be the set of users, n the number of advertisers, and k the number of slots. We use index u for users, i for advertisers and j for slots. At each point in time, a user $u \in U$ arrives. Each advertiser $i \in [n]$ bids a per-click value v_i^u on that user. This is the value the advertiser receives if the user clicks on their ad. Let $\mathtt{CTR}_{i,j}^u$ denote the click through rate of advertiser i in slot j, that is, the probability that the user u will click on the ad i if it is placed in slot j.

A truthful auction decides which ads to display in each of the k slots. The auction receives the vector $v = (v_1^u, \ldots, v_n^u)$ as well as the click through rates $\mathtt{CTR}_\cdot^u$ and returns an allocation matrix $\mathsf{a}(v) = [\mathsf{a}_{ij}]_{i\in[n],j\in[k]}$ where a_{ij} denotes the probability that ad i is displayed in slot j.[3] We omit the superscript u whenever it is clear from the context that we are discussing a certain user.

Truthfulness

Given an allocation $\mathsf{a}(v)$ (where the user u is implicit), advertiser i receives a net allocation (expected number of clicks) of $\sum_j \mathtt{CTR}_{i,j}^u \mathsf{a}_{ij}$ and a net expected value of $v_i \cdot \sum_j \mathtt{CTR}_{i,j}^u \mathsf{a}_{ij}$ from the allocation. To ensure truthfulness, there should exist a supporting pricing function $p_i(v)$ for every advertiser i such that bidding truthfully maximizes the advertiser's net expected utility. For such a payment function to exist, it is sufficient and necessary that the allocation probability $\sum_j \mathtt{CTR}_{i,j}^u \mathsf{a}_{ij}$ is monotone non-decreasing in the per-click value v_i. All of the mechanisms we discuss in this paper satisfy monotonicity. In Section 5 we discuss how to compute supporting payments efficiently.

Separable click through rates

Throughout this paper we assume that the click through rates $\mathtt{CTR}_{i,j}^u$ are multiplicatively separable into an advertiser-specific component and a slot-specific component. This is a standard model (see, for example, [1]).

▶ **Definition 1** (Separable Click Through Rates). *Click through rates are* separable *if, for every user* u*, there exists a advertiser dependent vector* $\alpha_u = (\alpha_1, \ldots, \alpha_n)$ *and a slot dependent vector* $\beta_u = (\beta_1, \ldots, \beta_k)$ *in which* $\alpha_1, \ldots, \alpha_n > 0$ *and* $1 \geq \beta_1 \geq \beta_2 \geq \ldots \geq \beta_k \geq 0$ *such that* $\mathtt{CTR}_{i,j}^u = \alpha_i \beta_j$ *for all* $i \in [n]$ *and* $j \in [k]$.

[3] We require $\sum_i \mathsf{a}_{ij} = 1$ for all j and $\sum_j \mathsf{a}_{ij} \leq 1$ for all i. Every matrix $\mathsf{a}(\cdot)$ satisfying these matching constraints can be expressed as a distribution over deterministic assignments of ads to slots.

Observe that in the separable model the value an advertiser i obtains from slot j is $\alpha_i\beta_j v_i$. Since the slot specific components β_j are common to all advertisers, the relative values of advertisers are given by $\alpha_i v_i$. These relative values are important in the mechanisms we design. We call them the "effective values" of the advertisers:

▶ **Definition 2** (Effective Value). *The effective value of advertiser* i *is given by* $\hat{v}_i = v_i\alpha_i$.

We call the above model of online advertising auctions with separable CTRs the **Position Auction Setting**.

Prior-free design

As in previous works, the mechanisms we design and analyze in this paper are prior-free, meaning that the allocation to a user does not depend on the distribution of users or advertisers' value vectors or the history of users already served. Besides the well-documented benefits of prior-free mechanism design, in the context of fairness we get the added benefit that fairness guarantees hold for all users that are served by the mechanism regardless of whether or not the auctioneer's model accounts for them.

▶ **Definition 3** (Scale-Free). *A mechanism is scale-free if it has the property that multiplying the input values by a uniform constant does not change the resulting allocation.*

2.1 Social Welfare

The goal of this work, as in [5, 4], is to achieve a tradeoff between fairness and social welfare for the mechanisms we design. The social welfare of an allocation $\mathsf{a}(v)$ is defined to be the sum of all of the advertisers' net expected values:

$$\mathrm{SW}(\mathsf{a}(v)) = \sum_{i\in[n],j\in[k]} v_i \mathtt{CTR}^{u}_{i,j}\mathsf{a}_{i,j}.$$

We compare this social welfare to the maximum achievable by any feasible allocation. When click through rates are separable, the maximum social welfare is achieved by the allocation that assigns advertisers to slots in decreasing order of $\hat{v}_i$, the effective values. We call the allocation sorted by effective values the Unfair-Opt and also use the same term to denote the social welfare of this allocation.

Formally, if π is the order of advertisers where $\hat{v}_{\pi_1} \geq \hat{v}_{\pi_2} \geq \ldots \geq \hat{v}_{\pi_n}$, then the (unfair) optimal social welfare is given by:

$$\textsc{Unfair-Opt}(v, \alpha, \beta) = \sum_{j=1}^{k} \alpha_{\pi_j} v_{\pi_j} \beta_j.$$

Since it is generally impossible to achieve optimal social welfare and fairness simultaneously, we look for mechanisms that guarantee our fairness notions while giving a good approximation to the optimal social welfare.

▶ **Definition 4** (Social Welfare Approximation). *We say mechanism* $\mathcal{A}(\cdot)$ *achieves an* η*-approximation to social welfare for* $\eta \leq 1$*, if for all instances* (v, α, β)*, we have* $\mathrm{SW}(\mathcal{A}(v, \alpha, \beta)) \geq \eta \cdot \textsc{Unfair-Opt}(v, \alpha, \beta)$.

2.2 Fairness

[5] formalized fairness in ad auctions as a value stability condition based on the notion of individual fairness. Individual fairness requires that the auction assign similar allocations to similar users. [5] defined similarity between two users on the basis of closeness between the value vectors assigned to them by the advertisers. Informally speaking, if two users receive similar values from all advertisers, then they should also receive similar allocations. In order for the definition to be scale-free with respect to values, similarity between values is defined in multiplicative terms.

In the context of a single item auction, allocations are probability vectors. Similarity in allocations is therefore defined based on some notion of distance between probability vectors. [5] formalized similarity in terms of the ℓ_∞ distance between the probability vectors whereas [4] used total variation or ℓ_1 distance. We state the value stability definition from [5] below.

▶ **Definition 5** (Definition 2.1 from [5], Value Stability). *An allocation mechanism* $\mathsf{a}(\cdot)$ *is value stable with respect to function* $\mathrm{f} : [1,\infty] \to [0,1]$ *if the following condition is satisfied for every pair of value vectors* v *and* v'*:*

$$|\mathsf{a}_\mathrm{i}(\mathrm{v}) - \mathsf{a}_\mathrm{i}(\mathrm{v}')| \le \mathrm{f}(\lambda) \text{ for all } \mathrm{i} \in [\mathrm{n}], \text{ where } \lambda = \max_{\mathrm{i}\in[\mathrm{n}]} \left(\max \left\{ \frac{\mathrm{v}_\mathrm{i}}{\mathrm{v}'_\mathrm{i}}, \frac{\mathrm{v}'_\mathrm{i}}{\mathrm{v}_\mathrm{i}} \right\} \right).$$

In this definition, the function f, called the value stability constraint, governs the strength of the value stability condition. We assume f to be non-decreasing, with $\mathrm{f}(0) = 0$ and $\mathrm{f}(\infty) = 1$. Following [5], we focus on the family of constraints $\mathrm{f}_\ell(\lambda) = 1 - \lambda^{-2\ell}$. [5] argue that this family of stability constraints captures the entire spectrum of possible fairness conditions in the context of allocation algorithms.

In order to extend these fairness definitions to the position auctions setting, we need to extend the notion of closeness in allocations to multi-dimensional allocation matrices M as well as extend the notion of closeness in values to click through rates.

Let us consider the latter issue first. A straightforward manner of extending closeness over value vectors to the separable setting is to require that two similar users are assigned similar values, as well as have similar click through rates. But this notion of closeness is too restrictive. Values capture how advertisers perceive users as potential customers; whereas click through rates capture how users perceive the relevance of ads to their needs and how users behave in perusing ads on a search page. Two users that are similarly qualified for a set of ads may nevertheless exhibit very different behavior in responding to ads on a search page. Ideally the fairness guarantees an allocation algorithm provides should hinge only on the closeness between values v_i and not on the closeness between click through rates $\mathtt{CTR}_{\mathrm{i},\mathrm{j}}$. However, in order to obtain good social welfare, allocations necessarily need to depend on the advertiser specific click through rates α_i. We accordingly define closeness between users in terms of their effective values $\alpha_\mathrm{i}\mathrm{v}_\mathrm{i}$ (while ignoring dissimilarity in slot specific CTRs, β). In Section 4 we extend our fairness definitions and guarantees to settings where closeness is defined only in terms of the values v_i, ignoring dissimilarity in α and β.

Let us now consider closeness over probability matrices. We consider three notions. The first is ℓ_∞ distance, the maximum difference of allocations in any one entry (i,j) of the corresponding matrices.

▶ **Definition 6** (Value Stability for Position Auctions). *An allocation mechanism* $\mathcal{A}(\cdot)$ *is value stable with respect to function* $\mathrm{f} : [1,\infty] \to [0,1]$ *if the following condition is satisfied for every set of value and CTR vectors* v*,* v'*,* α*,* α' *and* β*:*

$$|\mathrm{M}_{\mathrm{i},\mathrm{j}} - \mathrm{M}'_{\mathrm{i},\mathrm{j}}| \le 2\mathrm{f}_\ell(\lambda) \text{ for all } \mathrm{i} \in [\mathrm{n}], \mathrm{j} \in [\mathrm{k}] \text{ where } \lambda \text{ is defined as } \max_{\mathrm{i}\in[\mathrm{n}]} \left(\max \left\{ \frac{\alpha_\mathrm{i}\mathrm{v}_\mathrm{i}}{\alpha'_\mathrm{i}\mathrm{v}'_\mathrm{i}}, \frac{\alpha'_\mathrm{i}\mathrm{v}'_\mathrm{i}}{\alpha_\mathrm{i}\mathrm{v}_\mathrm{i}} \right\} \right)$$

and $\mathrm{M} = \mathcal{A}(\mathrm{v}, \alpha, \beta)$ *and* $\mathrm{M}' = \mathcal{A}(\mathrm{v}', \alpha', \beta)$.

Suppose, as an example, for a particular advertiser i, user u has an allocation of $a = (.1, .1, .1, .1)$. Consider two possible allocation vectors for some v close to u: $a' = (.15, .15, .15, .15)$ and $a'' = (.15, .05, .15, .05)$. In some sense, allocation a' is much more unfair than a'' because in a' the entry-wise differences from a compound while in a'' they offset each other. Weak value stability cannot distinguish these two cases because it is concerned only with the absolute differences. Our next definition, ordered value stability is intended to allow a'' but not a'.

To do this, we bound the absolute differences in the total allocation of an advertiser across all columns, weighted by a vector $h_{i,j}$. This vector represents the utility the first user receives from seeing advertisement i in slot j. Since we assume the slots are in decreasing order of salience, this should be weakly decreasing in j.

▶ **Definition 7** (Ordered Value Stability for Position Auctions). *An allocation mechanism $\mathcal{A}(\cdot)$ is ordered value stable with respect to function $f : [1, \infty] \to [0, 1]$ if the following condition is satisfied for every set of value and CTR vectors v, v', α, α' and β, as well as for any advertiser i and any decreasing vector h_i with $1 \geq h_{i,1} \geq \ldots \geq h_{i,k} \geq 0$:*

$$\left|\sum_{j=1}^{k} h_{i,j}\left(M_{i,j} - M'_{i,j}\right)\right| \leq f_\ell(\lambda) \text{ where } \lambda \text{ is defined as } \max_{i\in[n]}\left(\max\left\{\frac{\alpha_i v_i}{\alpha'_i v'_i}, \frac{\alpha'_i v'_i}{\alpha_i v_i}\right\}\right)$$

where $M = \mathcal{A}(v, \alpha, \beta)$ and $M' = \mathcal{A}(v', \alpha', \beta)$.

The previous two definitions are concerned only with a single advertiser. In some instances, however, there are meaningful subsets of advertisers and bounding the differences of the allocations each advertiser individually may not be sufficient to ensure fairness overall. For example, if there are several different ads giving information about registering to vote, the total volume of voter registration ads a user sees is more important from a fairness perspective than the amount they see any particular voter registration ad. Therefore, the last notion we consider is a combination of ℓ_1 and ℓ_∞ distance: we consider, for any subset of advertisers, the total variation distance between the allocations of these advertisers to one slot, and bound the maximum over all slots of this distance.

▶ **Definition 8** (Total Variation Value Stability for Position Auctions). *A mechanism $\mathcal{A}(\cdot)$ with satisfies total variation value stability with respect to a function $f : [1, \infty] \to [0, 1]$ if the following condition is satisfied for every set of value and CTR vectors v, v', α, α' and β, as well as every subset of advertisers $S \subseteq [n]$ and for every column j:*

$$\left|\sum_{s\in S} \mathcal{A}(\hat{v})_{s,j} - \sum_{s\in S} \mathcal{A}(\hat{v})_{s,j}\right| \leq f(\lambda) \text{ where } \lambda \text{ is defined as } \max_{i\in[n]}\left(\max\left\{\frac{\alpha_i v_i}{\alpha'_i v'_i}, \frac{\alpha'_i v'_i}{\alpha_i v_i}\right\}\right)$$

and where $M = \mathcal{A}(v, \alpha, \beta)$ and $M' = \mathcal{A}(v', \alpha', \beta)$.

3 Inverse Proportional Allocation

In this section, we present a generalization of the mechanism first introduced in [5] as IPA to the position auction setting. We show that the generalization retains a constant approximation to the optimal social welfare and an appropriate generalization of the value stability condition. In Section 3.1 we describe the generalization of the mechanism from $k = 1$ to general k. In Section 3.2 we show that two different value stability conditions hold and in Section 3.3 we show that the exact same guarantee in [5] holds for the generalization as well. Some of the proofs in this section are deferred to Appendix A.

3.1 Generalized IPA

In [5], IPA was presented as a mechanism for the single item auction. An interpretation of this mechanism is as follows: start with an infeasible allocation of 1 unit to each advertiser (for a total allocation of n) and then gradually decrease the allocations until the total allocation reaches 1. The rate of this decrease is determined by a function g of the reported values. The IPA with parameter ℓ uses $g(x) = x^{-\ell}$. [5] also presents an algorithmic interpretation of the mechanism. The following is the generalization of this mechanism to the position auctionsetting.

First, as a warm-up, we generalize IPA to a special case of the position auctionsetting where $\beta = \overrightarrow{1}$. Our algorithm allocates a total of k units to the advertisers, with each advertiser receiving an allocation $a_i \in [0, 1]$ such that $\sum_i a_i = k$.

We follow the same intuition as for the case of $k = 1$. The mechanism first allocates 1 to each advertiser, then decreases the allocations until the total allocation reaches k rather than 1. See Appendix A for an algorithmic interpretation of this mechanism. Note that setting $k = 1$ gives the exact same mechanism as in [5]. Algorithm 3 is scale free and produces allocations that are non-decreasing in k. Furthermore, the allocation to advertiser i, namely a_i, is non-decreasing in $\hat{v}_i$ and non-increasing in $\hat{v}_{-i}$.

We now extend the k-unit setting to the position auction setting. The resulting allocation algorithm is called Generalized IPA. The algorithm assigns to every slot j a distribution over advertisers given by the difference in the j-unit and $j-1$-unit allocations produced by k-unit IPA.

Feasibility

We observe that the allocation produced by the generalized IPA algorithm is feasible. That is, there exists a distribution over matchings from advertisers to slots, for which the total probability that advertiser i is allocated a slot is equal to M.

Algorithm 1 Generalized IPA.

Input: Vector v of non-negative advertiser bids for user u; CTRs $\alpha_1, \cdots, \alpha_n$ and $\beta_1, \cdots, \beta_k$; number of slots k; function $g : \mathbb{R}^{\geq 0} \to (0, \infty]$ with $g(0) = \infty$ and $\lim_{x\to\infty} g(x) = 0$;
for $h \in [k]$ **do**
  Set $\mathsf{a}^{(h)} \leftarrow$ the output of the IPA k-unit algorithm on input (v, α, h, g)
end
for $j \in [k]$ **do**
  Set $M_{\cdot,j} = \mathsf{a}^{(j)} - \mathsf{a}^{(j-1)}$
end
return M

Note that the generalized IPA algorithm is scale-free and independent of β.

3.2 Fairness

We now prove the value stability of the Generalized IPA mechanism.

▶ **Theorem 9.** *The Generalized IPA mechanism with parameter $\ell > 0$ and for any number of advertisers n is value stable with respect to any function f satisfying $f(\lambda) \geq f_\ell(\lambda) = 1 - \lambda^{-2\ell}$ for all $\lambda \in [1, \infty)$, as in Definition 6.*

Our proof has two parts. First, give a bound on the deviation between allocations given by the k-unit IPA mechanism to similar users. Then, we use the bound to show that Generalized IPA achieves value stability.

▶ **Lemma 10.** *For the* k*-unit IPA mechanism with parameter* ℓ *run on any* k *and any bid vectors* v *and* v' *with* $\lambda = \max_{\mathrm{i}\in[\mathrm{n}]}\{\hat{\mathrm{v}}_\mathrm{i}/\hat{\mathrm{v}}'_\mathrm{i}, \hat{\mathrm{v}}'_\mathrm{i}/\hat{\mathrm{v}}_\mathrm{i}\}$*, for all indices* i*,* $|\mathsf{a}_\mathrm{i}(\mathrm{v}) - \mathsf{a}_\mathrm{i}(\mathrm{v}')| \le \mathrm{f}_\ell(\lambda)$*.*

Next, we show that Generalized IPA satisfies ordered value stability.

▶ **Theorem 11.** *Generalized IPA with parameter* ℓ *satisfies ordered value stability with respect to* $\mathrm{f}_\ell(\lambda)$*. That is, for every set of value and CTR vectors* v*,* v'*,* α*,* α' *and* β*, as well as for any advertiser* i *and any decreasing vector* h *with* $1 \ge \mathrm{h}_1 \ge \ldots \ge \mathrm{h}_\mathrm{k} \ge 0$*:*

$$|\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{h}_\mathrm{j}\left(\mathrm{M}_{\mathrm{i},\mathrm{j}} - \mathrm{M}'_{\mathrm{i},\mathrm{j}}\right)| \le \mathrm{f}_\ell(\lambda) \text{ where } \lambda \text{ is defined as } \max_{\mathrm{i}\in[\mathrm{n}]}\left(\max\left\{\frac{\alpha_\mathrm{i}\mathrm{v}_\mathrm{i}}{\alpha'_\mathrm{i}\mathrm{v}'_\mathrm{i}}, \frac{\alpha'_\mathrm{i}\mathrm{v}'_\mathrm{i}}{\alpha_\mathrm{i}\mathrm{v}_\mathrm{i}}\right\}\right)$$

where $\mathrm{M} = \mathcal{A}(\mathrm{v}, \alpha, \beta)$ *and* $\mathrm{M}' = \mathcal{A}(\mathrm{v}', \alpha', \beta)$*.*

3.3 Social Welfare

We now show that Generalized IPA achieves a good approximation to the optimal social welfare Unfair-Opt.

▶ **Theorem 12.** *The IPA algorithm for the separable case, Algorithm 1, run with parameter* $\ell > 0$ *and any number of advertisers* n *achieves a* $\left(1 - \frac{\ell^\ell}{(1+\ell)^{\ell+1}}\right)$*-approximation the social welfare of the unfair optimum.*

To do so, we first show an approximation result for the special case of $\vec{\beta} = 1$, the k–unit algorithm.

▶ **Lemma 13.** *The IPA algorithm for the* k*–unit case, Algorithm 3, run with parameter* ℓ *and any number of advertisers* n *achieves a* $\left(1 - \frac{\ell^\ell}{(1+\ell)^{\ell+1}}\right)$*-approximation to the social welfare of the unfair optimum.*

We use Lemma 13 and extend definition of Generalized IPA allocation vector based on k–unit vectors to show Theorem 12. The proof is deferred to Appendix A. The approximation factor is $\frac{3}{4}$ at $\ell = 1$ and as $\ell \to \infty$, the approximation factor goes to 1.

▶ Remark 14. The approximation factor in Lemma 13 is tight for IPA mechanism.

Proof. Consider the following example. Fix a user u and let the bidding vector of the advertisers be:

$$(\underbrace{1, \ldots, 1}_{\mathrm{k}}, \overbrace{\epsilon, \ldots, \epsilon}^{\mathrm{n-k}})$$

where $1 > \epsilon = \frac{-5\mathrm{k} + \sqrt{25\mathrm{k}^2 - 16(\mathrm{n-k})\frac{\mathrm{k}^2}{\mathrm{n-k}} - 4 - 4(\mathrm{n-k})}}{8(\mathrm{n-k})} > 0$. Let $\ell = 1$ and $\mathrm{n} > 2\mathrm{k}$. We get:

$$\mathrm{SW}(\textsc{Alg}) = \mathrm{k}(1 - \frac{\mathrm{n}-\mathrm{k}}{(\mathrm{n}-\mathrm{k})\epsilon^{-1} + \mathrm{k}}) + (\mathrm{n}-\mathrm{k})\epsilon(1 - (\mathrm{n}-\mathrm{k})\frac{\epsilon^{-1}}{(\mathrm{n}-\mathrm{k})\epsilon^{-1} + \mathrm{k}}), \quad \textsc{Unfair-Opt} = \mathrm{k}.$$

For the aforementioned value of ϵ, we will have $\frac{\mathrm{SW}(\textsc{Alg})}{\textsc{Unfair-Opt}} = \frac{3}{4}$. Note that this example fits the maxima point we found in the proof of Lemma 13. ◀

4 Fairness for users with different preferences

So far we have assumed that similar users are similar in all aspects – the values advertisers assign to them as well as the rates at which the users click on different ads. However, these two sets of parameters are asymmetric. Values capture advertisers' preferences over users whereas CTRs capture users' preferences over advertisers. We will now distinguish between similarity in *qualification* (i.e. values) from similarity in *user preferences* (i.e. CTR).

A myopic viewpoint might suggest that two users that are similarly qualified should be treated similarly by the auction no matter their preferences. However, this is fundamentally at odds with the objective of maximizing the social welfare[4] a.k.a. the collective value of the advertisers, as the latter are contingent upon clicks. Consequently, the outcome of the auction cannot be completely independent of user preferences and we look towards a notion of fairness that is appropriately preference aligned.

To motivate our definitions, consider the following example. We have two users Alice and Bob, two advertisers A and B, and a single slot to display an ad. The users look identical to the advertisers: A places a value of \$1 on a click from either user and B places a value of \$10 from either click. However the users behave differently when they view ads. Bob clicks both ads with certainty. Alice clicks A's ad with certainty but B's ad with probability only 1%. The platform should clearly display ad A for Alice and ad B for Bob. Although these outcomes are different, both users are happy: Bob is essentially indifferent between A and B, while Alice greatly prefers A. In this case, any differences in allocation are aligned with user preferences.

Can we always expect this to be the case? Formally, consider a single slot auction with n advertisers, and two users with identical value vectors $\mathrm{v} = \mathrm{v}'$. Let a and a' denote their respective allocation vectors. Can we ensure that any allocation mass that is moved between advertisers in a' relative to a is moved from low CTR advertisers to high CTR advertisers?

Unfortunately, we cannot ensure this property while also maintaining a reasonable approximation for social welfare. To see this, consider the above example with Alice and Bob once again and suppose that Bob's CTR for advertiser B changes to 20%. In order to obtain a good social welfare, the auction must continue to display ad B for Bob. However, now Bob gets to see much more of ad B and much less of ad A than Alice even though he greatly prefers ad A to ad B. The key observation here is that the allocation mass in B's allocation shifts to an advertiser with high *relative* CTR, when measured relative to the CTRs of Alice.

Motivated by this example, we propose the following new preference-aligned definition of fairness for identically valued users. Underlying this definition is a relative ordering of advertisers for two users u and v with advertiser specific CTR vectors $\alpha_{\mathrm{u}} = (\alpha^{\mathrm{u}}_1, \cdots, \alpha^{\mathrm{u}}_{\mathrm{n}})$ and $\alpha_{\mathrm{v}} = (\alpha^{\mathrm{v}}_1, \cdots, \alpha^{\mathrm{v}}_{\mathrm{n}})$. We will assume that advertisers are ordered in (weakly) decreasing order of the ratio $\alpha^{\mathrm{v}}_{\mathrm{i}}/\alpha^{\mathrm{u}}_{\mathrm{i}}$, and require that allocation mass for user v is shifted from advertisers that appear later in the ordering to those that appear earlier in the ordering.

▶ **Definition 15** (Value Stability for Identically-Valued Users with Heterogeneous Preferences). *An allocation mechanism $\mathcal{A}(\cdot)$ is value-stable for identical users with heterogeneous preferences if for every pair of users with identical value vectors* v*; CTR vectors α, α', β, and β'; any ordering over advertisers that is weakly decreasing in α/α'; and for every advertiser* $\mathrm{i} \in [\mathrm{n}]$ *and slot* $\mathrm{j} \in [\mathrm{k}]$*:*

$$\sum_{\mathrm{t}=1}^{\mathrm{i}}\sum_{\mathrm{s}=1}^{\mathrm{j}} \mathrm{M}_{\mathrm{t,s}} \geq \sum_{\mathrm{t}=1}^{\mathrm{i}}\sum_{\mathrm{s}=1}^{\mathrm{j}} \mathrm{M}'_{\mathrm{t,s}}, \qquad \textit{where } \mathrm{M} = \mathcal{A}(\mathrm{v}, \alpha, \beta) \textit{ and } \mathrm{M}' = \mathcal{A}(\mathrm{v}, \alpha', \beta').$$

[4] Social welfare is a misnomer in this context, as it does not take into account the benefit or value users derive from viewing the ad.

Similar users

The above definition extends in a straightforward manner to pairs of users that are similarly rather than identically qualified, and again have different preferences over advertisers as expressed through CTRs. Once again we require that allocation mass shifts from advertisers with low relative CTR to those with higher relative CTR, but we allow for additive errors in allocation that grow with the dissimilarity in the users' values.

▶ **Definition 16** (Value Stability for Similarly-Valued Users with Heterogeneous Preferences)**.** *An allocation mechanism* $\mathcal{A}(\cdot)$ *is value-stable for users with heterogeneous preferences with respect to function* $\mathrm{f}_\ell : [1,\infty] \to [0,1]$ *if for every pair of users with value vectors* v *and* v′*; CTR vectors* α*,* α'*,* β*, and* β'*; any ordering over advertisers that is weakly decreasing in* α/α'*; and for every advertiser* $\mathrm{i} \in [\mathrm{n}]$ *and slot* $\mathrm{j} \in [\mathrm{k}]$*:*

$$\sum_{\mathrm{t}=1}^{\mathrm{i}}\sum_{\mathrm{s}=1}^{\mathrm{j}} \mathrm{M}_{\mathrm{t,s}} \geq \sum_{\mathrm{t}=1}^{\mathrm{i}}\sum_{\mathrm{s}=1}^{\mathrm{j}} \mathrm{M}'_{\mathrm{t,s}} - \mathrm{if}_\ell(\lambda)$$

where $\mathrm{M} = \mathcal{A}(\mathrm{v}, \alpha, \beta)$, $\mathrm{M}' = \mathcal{A}(\mathrm{v}', \alpha', \beta')$ *and* $\lambda = \max_{\mathrm{i}\in[\mathrm{n}]} \left\{ \max\left\{ \frac{\mathrm{v}_\mathrm{j}}{\mathrm{v}_\mathrm{i}}, \frac{\mathrm{v}'_\mathrm{i}}{\mathrm{v}_\mathrm{i}} \right\} \right\}$.

Comparing Definition 15 and Definition 16, note that if $\mathrm{v} = \mathrm{v}'$ then $\lambda = 1$ and, as discussed in [5], a proper f function has the property of $\mathrm{f}(1) = 0$. Therefore, Definition 15 is exactly Definition 16 in the special case of $\mathrm{v} = \mathrm{v}'$.

4.1 Fairness of IPA and PA for heterogeneous users

We show that both the Generalized IPA and Generalized PA mechanisms satisfy Definition 15 and more generally Definition 16.

To begin, we show that any mechanism for the k-unit case satisfying certain mild conditions also satisfies Definition 15. Both k-unit IPA and k-unit PA satisfy these conditions and hence are value-stable for identically qualified users with heterogeneous preferences.

▶ **Lemma 17.** *Let* $\mathsf{a}(\mathrm{v})$ *be a scale-free* k*-unit allocation algorithm such that* $\mathsf{a}_\mathrm{i}(\mathrm{v})$ *is weakly increasing in* v_i*. Suppose further that for all* $\mathrm{t} \neq \mathrm{i}$*,* $\mathsf{a}_\mathrm{i}(\mathrm{v})$ *is weakly decreasing in* v_t*. Then* $\mathsf{a}(\mathrm{v})$ *satisfies Definition 15.*

Proof. Fix i and scale α' so that $\alpha_\mathrm{i} = \alpha'_\mathrm{i}$. Since the advertisers are sorted, we now know that for all $\mathrm{t} < \mathrm{i}$, $\alpha_\mathrm{t} \geq \alpha'_\mathrm{t}$ and for all $\mathrm{t} > \mathrm{i}$, $\alpha_\mathrm{t} \leq \alpha'_\mathrm{t}$.

We proceed by two cases and then use a transitivity argument to show the theorem holds in general.

Consider the case where for all $\mathrm{t} \leq \mathrm{i}$, $\alpha_\mathrm{t} = \alpha'_\mathrm{t}$. Then $\alpha\mathrm{v} \begin{cases} = \alpha'\mathrm{v} \text{ for all } \mathrm{t} \leq \mathrm{i} \\ \leq \alpha'\mathrm{v} \text{ for all } \mathrm{t} > \mathrm{i} \end{cases}$.

Therefore, since the allocation a_t is weakly decreasing in v_s for all $\mathrm{s} \neq \mathrm{t}$, we have that for all $\mathrm{t} \leq \mathrm{i}$, $\mathrm{a}(\alpha\mathrm{v}) \geq \mathrm{a}(\alpha'\mathrm{v})$. Hence, $\sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v}) \geq \sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v})$, as desired.

Now, consider the case where for all $\mathrm{t} \geq \mathrm{i}$, $\alpha_\mathrm{t} = \alpha'_\mathrm{t}$. Then $\alpha\mathrm{v} \begin{cases} \geq \alpha'\mathrm{v} \text{ for all } \mathrm{t} < \mathrm{i} \\ = \alpha'\mathrm{v} \text{ for all } \mathrm{t} \geq \mathrm{i} \end{cases}$.

Therefore, since the allocation a_t is weakly decreasing in v_s for all $\mathrm{s} \neq \mathrm{t}$, we have that for all $\mathrm{t} > \mathrm{i}$, $\mathrm{a}(\alpha\mathrm{v}) \leq \mathrm{a}(\alpha'\mathrm{v})$ and hence $\sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v}) \leq \sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v})$. But $\sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v}) = \mathrm{k} - \sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v})$ and likewise $\sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v}) = \mathrm{k} - \sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v})$. Therefore, $\sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v}) \leq \sum_{\mathrm{t}=\mathrm{i}+1}^{\mathrm{n}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v})$ implies $\sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha\mathrm{v}) \geq \sum_{\mathrm{t}=1}^{\mathrm{i}} \mathrm{a}_\mathrm{t}(\alpha'\mathrm{v})$, as desired.

We now argue that the theorem holds in general. Let $\alpha''_t := \begin{cases} \alpha_t \text{ if } t \le i \\ \alpha'_t \text{ if } t > i \end{cases}$. By the first case, $\sum_{t=1}^{i} a_t(\alpha v) \ge \sum_{t=1}^{i} a_t(\alpha'' v)$, and by the second case $\sum_{t=1}^{i} a_t(\alpha'' v) \ge \sum_{t=1}^{i} a_t(\alpha' v)$. Hence, $\sum_{t=1}^{i} a_t(\alpha v) \ge \sum_{t=1}^{i} a_t(\alpha' v)$, as desired. ◀

▶ **Corollary 18.** *The* k*-unit IPA and* k*-unit PA mechanisms satisfy Definition 15.*

Because our generalized mechanisms are defined in terms of telescoping differences of the k-unit allocations, Theorem 19 follows directly from Corollary 18.

▶ **Theorem 19.** *The Generalized IPA and Generalized PA mechanisms satisfy Definition 15.*

Next, we show Generalized IPA and Generalized PA are value-stable for similarly-valued users with heterogeneous preferences. The only thing changing from Definition 15 to Definition 16 is that we need to keep track of small changes between the two allocations, which leads to the following theorem. The proof is deferred to Appendix B.

▶ **Theorem 20.** *The Generalized IPA and Generalized PA mechanisms $\mathcal{A}(\cdot)$ with parameter ℓ are value-stable for similarly-valued users with heterogeneous preferences.*

5 Computing payments

In this section we develop an algorithm for computing supporting payments for the generalized IPA and generalized PA allocation rules. Our main observation is that the allocation functions of IPA and PA are piecewise rational functions with polynomially many pieces where each piece can be computed in polynomial time. With these pieces in hand, and using Myerson's lemma, computing payments amounts to computing polynomially many integrals of rational functions.

We focus on the generalized IPA; the argument for generalized PA is similar. Formally, for a fixed and implicit user u, and a fixed and implicit advertiser i, let $x_i(v)$ denote the net allocation to the advertiser, a.k.a. the expected number of clicks the advertiser receives from the user. If the user is assigned allocation $M = \mathcal{A}(v, \alpha, \beta)$ then we have $x_i(v) = \sum_j M_{i,j}\alpha_i\beta_j$. Let $a^{(j)}$ denote the cumulative allocation to the user in the first j slots as in the description of Algorithm 2 and recall that $M_{i,j} = a_i^{(j)} - a_i^{(j-1)}$. Accordingly we get:

$$x_i(v) = \alpha_i \sum_j a_i^{(j)}(\beta_j - \beta_{j+1}). \tag{1}$$

In other words, $x_i(v)$ is a linear combination of the functions $a_i^{(j)}(v)$.

We will now argue that for all i, j, the function $a_i^{(j)}(v)$, as defined in Algorithm 1, is piecewise rational in v_i. Consider the following equivalent formulation of Algorithm 1. Given the values $v_1, \cdots, v_n$, ad-specific CTRs $\alpha_1, \alpha_2, \cdots, \alpha_n$, and decreasing function g, we find a parameter t such that

$$\sum_{i'} \min(1, t \cdot g(\alpha_{i'} v_{i'})) = n - j. \tag{2}$$

The allocation $a_i^{(j)}$ is then given by $1 - \min(1, t \cdot g(\alpha_i v_i))$.

Suppose without loss of generality that i receives a non-zero allocation at value v_i (otherwise $a_i^{(j)}$ is trivially piecewise rational at values $\le v_i$). We can then rewrite Equation (2) as:

$$t \cdot g(\alpha_i v_i) + \sum_{i' \ne i} \min(1, t \cdot g(\alpha_{i'} v_{i'})) = n - j. \tag{3}$$

Now, the expression $\sum_{i' \neq i} \min(1, tg(\alpha_{i'} v_{i'}))$ is independent of v_i and piecewise linear in t with at most n pieces. Given the values v_{-i} and CTRs α_{-i}, we can efficiently compute the linear pieces in this function. Substituting any particular linear piece with t in the range $[t_1, t_2]$ in Equation (3) then gives us an equation of the following form with appropriate parameters x and y:

$$t \cdot g(\alpha_i v_i) + xt = y$$

leading to the solution

$$a_i^{(j)}(v_i) = 1 - g(\alpha_i v_i) \cdot \frac{y}{g(\alpha_i v_i) + x} \quad \text{for } v_i \in \left[\frac{1}{\alpha_i} g^{-1}\left(\frac{y - xt_2}{t_2}\right), \frac{1}{\alpha_i} g^{-1}\left(\frac{y - xt_1}{t_1}\right) \right].$$

Observe that the RHS in the above equation is a rational function as the function g in the definition of IPA is also rational.

Summarizing, we first compute the piecewise rational form of the function $a_i^{(j)}(v_i)$ for all slots j. Each of these functions has at most n pieces. We then use Equation (1) to express $x_i(v_i)$ as a piecewise rational function with at most nk pieces. Finally, we use Myerson's lemma and compute per-impression payments as

$$p_i(v_i) = v_i x_i(v_i) - \int_{z=0}^{v_i} x_i(z)\, dz.$$

6 Proportional Allocation

In this section, we present a generalization of the mechanism first introduced in [4] as Proportional Allocation (PA) to the position auction setting. We show that the generalization retains the same approximation ratio to the optimal social welfare and an appropriate generalization of the total variation value stability condition. This is a stronger fairness guarantee than that of Generalized IPA, but comes at the cost of a weaker approximation to the optimal social welfare. For a detailed discussion of the trade-offs between the single-unit versions these methods, see [5]. Some of the proofs in this section are deferred to Appendix C.

6.1 Generalized PA

In contrast to IPA, PA can be thought of as initially assigning each advertiser an allocation of 0 and then increasing the allocations in proportion to (some function of) the bid amounts until the total allocation reaches 1. [4] analyzes this mechanism for the single unit case. In particular, they prove value stability with respect to the total variation distance on the allocations, rather than with respect to the ℓ^∞ distance as with IPA. However, in exchange, the social welfare approximation achieved by PA degrades as the number of advertisers increases.

Just like the previous section, we start with a warm-up case in which we consider a special case of position auctionwhere $\beta = \overrightarrow{1}$. For this case, we will attempt to allocate proportionally, assigning $k \cdot \frac{g(v_i)}{\sum_t g(v_t)}$ to each bidder i. If this allocation is more than 1 for any advertiser, we cap their allocation at 1 and divide the additional mass proportionally among the remaining advertisers. See Algorithm 4 in Appendix C for an algorithmic interpretation of this mechanism. Note that the function g in this mechanism is different than the one in Section 3, as it is a continuous, super-additive and increasing function.

The extension of this algorithm to the position auctioncase is similar to the extension we saw in Section 3 for IPA, and works as follows:

Algorithm 2 Generalized PA.

Input: Vector v of non-negative advertiser bids for user u; CTRs $\alpha_1, \cdots, \alpha_n$ and $\beta_1, \cdots, \beta_k$; number of slots k; function $g : \mathbb{R}^{\geq 0} \to [0, \infty]$ with g a continuous, super-additive, increasing function and $g(0) = 0$;
for $h \in [k]$ **do**
  Set $p^{(h)} \leftarrow$ the output of the PA k-unit algorithm on input (v, α, h, g)
end
for $j \in [k]$ **do**
  Set $P_{\cdot,j} = p^{(j)} - p^{(j-1)}$
end
return P

Observe that Generalized PA is scale-free, independent of β, and produces feasible allocations.

6.2 Fairness

First, we prove the fairness guarantees of our mechanism. We begin by showing the total variation value stability of PA, which as we've discussed is the main advantage of PA over IPA.

▶ **Theorem 21.** *The Generalized PA mechanism with parameter* $g(x) = x^\ell$ *satisfies Definition 8 Total Variation Value Stability for Position Auctions with respect to* $f_\ell(\lambda)$*. That is, for all pairs of effective value vectors* $\hat{v}, \hat{v}'$*, subsets of advertisers* $S \subseteq [n]$*, and slots* j*,*

$$|\sum_{s \in S} P_{s,j}(\hat{v}) - \sum_{s \in S} P_{s,j}(\hat{v}')| \leq 2 f_\ell(\lambda).$$

The proof of Theorem 21 uses the following key lemma, which shows a similar property holds for k-unit PA mechanism.

▶ **Lemma 22.** *The* k*-unit PA mechanism with parameter* $g(x) = x^\ell$ *satisfies the property that, for all pairs of effective value vectors* $\hat{v}, \hat{v}'$ *and subsets of advertisers* $S \subseteq [n]$*,*

$$|\sum_{s \in S} a_s(\hat{v}) - \sum_{s \in S} a_s(\hat{v}')| \leq \frac{\lambda^\ell - 1}{\lambda^\ell + 1} \leq f_\ell(\lambda).$$

We now show that Generalized PA also satisfies the same ordered value stability property as IPA. The proof is essentially identical as the proof of Theorem 11 except in that it uses the total variation value stability of PA instead the value stability of IPA. For the full proof, see Appendix C.

▶ **Theorem 23.** *Generalized PA with parameter* ℓ *satisfies ordered value stability with respect to* $f_\ell(\lambda)$*. That is, for every set of value and CTR vectors* v*,* v'*,* α*,* α' *and* β*, as well as for any advertiser* i *and any decreasing vector* h *with* $1 \geq h_1 \geq \ldots \geq h_k \geq 0$*:*

$$|\sum_{j=1}^{k} h_j \left(P_{i,j} - P'_{i,j}\right)| \leq f_\ell(\lambda) \text{ where } \lambda \text{ is defined as } \max_{i \in [n]} \left(\max \left\{ \frac{\alpha_i v_i}{\alpha'_i v'_i}, \frac{\alpha'_i v'_i}{\alpha_i v_i} \right\} \right)$$

where $P = \mathcal{A}(v, \alpha, \beta)$ *and* $P' = \mathcal{A}(v', \alpha', \beta)$*.*

6.3 Social Welfare

Finally, we give our guarantee on the social welfare approximation ratio achieved by Generalized PA relative to Unfair-Opt. The proof relies on a lemma showing the same approximation result for the special case of $\vec{\beta} = 1$, k-unit PA.

▶ **Theorem 24.** *The Generalized PA mechanism with parameter ℓ achieves a $\left(\frac{n-k}{n}(n-k)^{-1/\ell} + 1/n\right)$-approximation to the optimal social welfare for any instance with* n *advertisers and* k *slots.*

▶ **Lemma 25.** *The* k*-unit PA subroutine with parameter ℓ achieves a $\left(\frac{n-k}{n}(n-k)^{-1/\ell} + 1/n\right)$-approximation to the optimal social welfare for any instance with* n *advertisers and* k *slots.*

References

1 Gagan Aggarwal, Ashish Goel, and Rajeev Motwani. Truthful auctions for pricing search keywords. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 1–7, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1134707.1134708.

2 Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. Discrimination through optimization: How facebook's ad delivery can lead to biased outcomes. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), November 2019. doi:10.1145/3359301.

3 Andrea Celli, Riccardo Colini-Baldeschi, and Stefano Leonardi. Learning fair equilibria in sponsored search auctions. *arXiv preprint*, 2021. arXiv:2107.08271.

4 Shuchi Chawla, Christina Ilvento, and Meena Jagadeesan. Multi-category fairness in sponsored search auctions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 348–358, New York, NY, USA, 2020. Association for Computing Machinery.

5 Shuchi Chawla and Meena Jagadeesan. Individual fairness in advertising auctions through inverse proportionality. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

6 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*, ITCS '12, pages 214–226, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090255.

7 Cynthia Dwork and Christina Ilvento. Fairness under composition. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 33:1–33:20, 2019.

8 Meryem Essaidi and S Matthew Weinberg. On symmetries in multi-dimensional mechanism design. In *International Conference on Web and Internet Economics*, pages 59–75. Springer, 2021.

9 Lodewijk Gelauff, Ashish Goel, Kamesh Munagala, and Sravya Yandamuri. Advertising for demographically fair outcomes. *arXiv preprint*, 2020. arXiv:2006.03983.

10 Julia Angwin and Terry Perris. Facebook lets advertisers exclude users by race, 2016. URL: https://www.propublica.org/article/facebook-lets-advertisers-exclude-users-by-race.

11 Julia Angwin, Noam Scheiber, and Ariana Tobin. Facebook job ads raise concerns about age discrimination, 2017. URL: https://www.nytimes.com/2017/12/20/business/facebook-job-ads.html.

12 Katie Benner, Glenn Thrush, and Mike Isaac. Facebook engages in housing discrimination with its ad practices, u.s. says, 2019. URL: https://www.nytimes.com/2019/03/28/us/politics/facebook-housing-discrimination.html.

13 Michael P. Kim, Aleksandra Korolova, Guy N. Rothblum, and Gal Yona. Preference-informed fairness. In *Proceedings of the 11th Innovations in Theoretical Computer Science, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020*, page to appear, 2020.

14 A Lambrecht and C E Tucker. Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management Science*, 65(7):2966–2981, July 2019. © 2019 INFORMS This manuscript has been accepted for publication in Management Science. The version of record can be found at `doi:10.1287/mnsc.2018.3093`. URL: `https://lbsresearch.london.edu/id/eprint/967/`.

15 Milad Nasr and Michael Carl Tschantz. Bidding strategies with gender nondiscrimination constraints for online ad auctions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 337–347, New York, NY, USA, 2020. Association for Computing Machinery. `doi:10.1145/3351095.3375783`.

16 Alison Watts. Fairness and efficiency in online advertising mechanisms. *Games*, 12(2):36, 2021.

17 Di Yuan, Manmohan Aseri, and Tridas Mukhopadhyay. Is fair advertising good for platforms? *Available at SSRN 3913739*, 2021.

18 Muhammad Bilal Zafar, Isabel Valera, Manuel Rodriguez, Krishna Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. *Advances in Neural Information Processing Systems*, 30, 2017.

A Deferred Proofs from Section 3

Below is the algorithmic description of position auctionin the case of $\vec{\beta} = 1$:

Algorithm 3 k-unit IPA.

Input: Vector v of non-negative advertiser bids for user u; ad-specific CTRs $\alpha_1, \cdots, \alpha_n$; number of slots k; function $g : \mathbb{R}^{\geq 0} \to (0, \infty]$ with $g(0) = \infty$ and $\lim_{x\to\infty} g(x) = 0$;
Initialization: Determine effective values, $\hat{v}_i = v_i\alpha_i$ for all i;
WLOG assume $\hat{v}_1 \geq \ldots \geq \hat{v}_n$;
if $k = 0$ **then**
  return $a(v) = \overrightarrow{0}$
end
if $\hat{v}_1 \leq 0$ **then**
  Set $a_i = \frac{k}{n}$ for all $i \in [n]$, return $a(v)$;
end
Set $s \leftarrow \max\{i \in [n] : \hat{v}_i > 0\}$;
while $(s-k)g(\hat{v}_s) \geq \sum_{i=1}^{s} g(\hat{v}_i)$ **do**
  $s \leftarrow s - 1$;
end
For $i > s$: set $a_i = 0$;
For $i \leq s$ set $a_i = 1 - (s-k)\frac{g(\hat{v}_i)}{\sum_{t=1}^{s} g(\hat{v}_t)}$;
return $a(v)$

▶ **Theorem 11.** Generalized IPA with parameter ℓ satisfies ordered value stability with respect to $f_\ell(\lambda)$. That is, for every set of value and CTR vectors v, v′, α, α' and β, as well as for any advertiser i and any decreasing vector h with $1 \geq h_1 \geq \ldots \geq h_k \geq 0$:

$$\left|\sum_{j=1}^{k} h_j \left(M_{i,j} - M'_{i,j}\right)\right| \leq f_\ell(\lambda) \text{ where } \lambda \text{ is defined as } \max_{i\in[n]} \left(\max\left\{\frac{\alpha_i v_i}{\alpha'_i v'_i}, \frac{\alpha'_i v'_i}{\alpha_i v_i}\right\}\right)$$

where $M = \mathcal{A}(v, \alpha, \beta)$ and $M' = \mathcal{A}(v', \alpha', \beta)$.

Proof. Fix some vectors v, v', α, α', and β, and the corresponding allocation matrices M and M'. Consider some advertiser i. We begin by using the definition Generalized IPA and then rearranging terms. Note that we define $h_{k+1} := 0$ for notational simplicity.

$$\begin{aligned}
|\sum_{j=1}^{k} h_j \left(M_{i,j} - M'_{i,j}\right)| &= |\sum_{j=1}^{k} h_j \left((a_i^{(j)} - a_i^{(j-1)}) - (a_{i'}^{(j)} - a_{i'}^{(j-1)})\right)| \\
&= |\sum_{j=1}^{k} \left(h_j(a_i^{(j)} - a_i^{(j-1)}) - h_j(a_{i'}^{(j)} - a_{i'}^{(j-1)})\right)| \\
&= |\sum_{j=1}^{k} \left(a_i^{(j)} - a_{i'}^{(j-1)}\right)\left(h_j - h_{j+1}\right)|.
\end{aligned}$$

Now, observe that because $h_1 \leq 1$ and the coefficients $(h_j - h_{j+1})$ telescope, the sum of these coefficients is at most 1. Since the expression is a weighted sum over columns of the differences in allocation at that column, the expression is bounded by the maximum difference in any column. But because Generalized IPA satisfies value stability (by Lemma 10), this is bounded by $f_\ell(\lambda)$, as desired.

$$|\sum_{j=1}^{k} h_j \left(M_{i,j} - M'_{i,j}\right)| = |\sum_{j=1}^{k} \left(a_i^{(j)} - a_{i'}^{(j-1)}\right)\left(h_j - h_{j+1}\right)| = |\max_j \left(a_i^{(j)} - a_{i'}^{(j-1)}\right)| \leq f_\ell(\lambda) \quad \blacktriangleleft$$

▶ **Theorem 12.** The IPA algorithm for the separable case, Algorithm 1, run with parameter $\ell > 0$ and any number of advertisers n achieves a $\left(1 - \frac{\ell^\ell}{(1+\ell)^{\ell+1}}\right)$-approximation the social welfare of the unfair optimum.

Proof. Suppose the k-unit IPA mechanism attains an η approximation to the optimal social welfare in the k-unit setting. Then the Generalized IPA mechanism attains the same approximation factor η in the position auctionwhen run with the k-unit IPAmechanism as a subroutine. In order to prove this, we consider the social welfare attained by the Generalized IPA mechanism. Since $\beta_{k+1} = 0$ and $a_i^{(0)} = \vec{0}$,

$$\text{SW}(\textsc{Alg}) = \sum_{i=1}^{n}\sum_{j=1}^{k} \alpha_i v_i \beta_j M_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{k} \hat{v}_i \beta_j \left[a_i^{(j)} - a_i^{(j-1)}\right] = \sum_{i=1}^{n}\sum_{j=1}^{k} \hat{v}_i (\beta_j - \beta_{j+1}) a_i^{(j)}.$$

Since for all $j \in [k]$, $\sum_i \hat{v}_i a_i^{(j)} \geq \eta(\hat{v}_1 + \cdots + \hat{v}_j)$, then:

$$\begin{aligned}
\text{SW}(\textsc{Alg}) &= \sum_{j=1}^{k} (\beta_j - \beta_{j+1}) \left(\sum_{i=1}^{n} \hat{v}_i a_i^{(j)}\right) \geq \eta \sum_{j=1}^{k} (\beta_j - \beta_{j+1}) \left(\hat{v}_1 + \cdots + \hat{v}_j\right) \\
&= \eta \sum_{j=1}^{k} \hat{v}_j \beta_j = \eta \, \textsc{Unfair-Opt}.
\end{aligned}$$

Finally, we know by Lemma 13 that the k-unit IPA mechanism is an $\eta = \left(1 - \frac{\ell^\ell}{(1+\ell)^{\ell+1}}\right)$-approx -imation to the optimal k-unit social welfare. Replacing η by $\left(1 - \frac{\ell^\ell}{(1+\ell)^{\ell+1}}\right)$ concludes the proof. ◀

B Deferred Proofs from Section 4

▶ **Theorem 20.** The Generalized IPA and Generalized PA mechanisms $\mathcal{A}(\cdot)$ with parameter ℓ are value-stable for similar users with heterogeneous preferences.

Proof. Fix users with user-dependent CTR vectors α and α' and value vectors v and v′. Also fix slot-dependent CTR vector β, advertiser i, and column j. Let $M = \mathcal{A}(v, \alpha, \beta)$ and $M' = \mathcal{A}(v, \alpha', \beta)$, where $a_t = \sum_{s=1}^{j} M_{t,s}$, and $a'_t = \sum_{s=1}^{j} M'_{t,s}$. Finally, fix a permutation π on advertisers for which $\frac{\alpha_{\pi_1}}{\alpha'_{\pi_1}} \geq \ldots \geq \frac{\alpha_{\pi_n}}{\alpha'_{\pi_n}}$.

Since $\mathcal{A}(\cdot)$ is envy-free, we know that $\sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha v) \geq \sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha' v)$. Therefore, it suffices to show $\sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha' v) \geq \sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha' v') - if(\lambda)$. Consider the difference $\sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha' v') - \sum_{s=1}^{i}\sum_{t=1}^{j} M_{st}(\alpha' v)$. Since $\sum_{t=1}^{j} M_{st}(\alpha v) = a_s^j(\alpha v)$, we can simplify this to:

$$\sum_{s=1}^{i} a_s^j(\alpha' v') - \sum_{s=1}^{i} a_s^j(\alpha' v) \leq |\sum_{s=1}^{i} a_s^j(\alpha' v') - \sum_{s=1}^{i} a_s^j(\alpha' v)| = |\sum_{s=1}^{i} a_s^j(\alpha' v') - a_s^j(\alpha' v)|$$
$$\leq \sum_{s=1}^{i} |a_s^j(\alpha' v') - a_s^j(\alpha' v)| \leq \sum_{s=1}^{i} f(\lambda) = i * f(\lambda).$$

Simply combining this with the previous inequality gives the desired result. ◀

C Deferred Proofs from Section 6

Below is the algorithmic description of position auctionin the case of $\vec{\beta} = 1$:

Algorithm 4 k-unit PA.

Input: Vector v of non-negative advertiser bids for user u; ad-specific CTRs $\alpha_1, \cdots, \alpha_n$; number of slots k; function $g : \mathbb{R}^{\geq 0} \to [0, \infty]$ with g a continuous, super-additive, increasing function and $g(0) = 0$;
Initialization: Determine effective values, WLOG assume $\hat{v}_1 \geq \ldots \geq \hat{v}_n$;
if $k = 0$ **then**
  return $p(v) = \overrightarrow{0}$
end
if $\hat{v}_1 \leq 0$ **then**
  Set $p_i = \frac{k}{n}$ for all $i \in [n]$, return p(v);
end
Set $s \leftarrow \max\{i \in [n] : \hat{v}_i > 0\}$;
Set $r = 1$;
while $\frac{k.g(\hat{v}_r)}{\sum_{t=r}^{s} g(\hat{v}_t)} \geq 1$ **do**
  $p_r = 1$;
  $r \leftarrow r + 1$;
end
For $i \geq r$: set $p_i = \frac{(k-r).g(\hat{v}_i)}{\sum_{t=r}^{s} g(\hat{v}_t)}$;
return p(v);

▶ **Theorem 23.** Generalized PA with parameter ℓ satisfies ordered value stability with respect to $f_\ell(\lambda)$. That is, for every set of value and CTR vectors v, v', α, α' and β, as well as for any advertiser i and any decreasing vector h with $1 \geq h_1 \geq \ldots \geq h_k \geq 0$:

$$|\sum_{j=1}^{k} h_j \left(P_{i,j} - P'_{i,j}\right)| \leq f_\ell(\lambda) \text{ where } \lambda \text{ is defined as } \max_{i\in[n]} \left(\max\left\{\frac{\alpha_i v_i}{\alpha'_i v'_i}, \frac{\alpha'_i v'_i}{\alpha_i v_i}\right\}\right)$$

where $P = \mathcal{A}(v, \alpha, \beta)$ and $P' = \mathcal{A}(v', \alpha', \beta)$.

Proof. Fix some vectors v, v', α, α', and β, and the corresponding allocation matrices P and P'. Consider some advertiser i. We begin by using the definition of Generalized PA and then rearranging terms. Note that we define $h_{k+1} := 0$ for notational simplicity.

$$\begin{aligned}
|\sum_{j=1}^{k} h_j \left(P_{i,j} - P_{i,j}\right)| &= |\sum_{j=1}^{k} h_j \left((p_i^{(j)}(\hat{v}) - p_i^{(j-1)}(\hat{v})) - (p_i^{(j)}(\hat{v}') - p_i^{(j-1)}(\hat{v}'))\right)| \\
&= |\sum_{j=1}^{k} \left(h_j(p_i^{(j)}(\hat{v}) - p_i^{(j-1)}(\hat{v})) - h_j(p_i^{(j)}(\hat{v}') - p_i^{(j-1)}(\hat{v}'))\right)| \\
&= |\sum_{j=1}^{k} \left(p_i^{(j)}(\hat{v}) - p_i^{(j-1)}(\hat{v}')\right) \left(h_j - h_{j+1}\right)|.
\end{aligned}$$

Now, observe that because $h_1 \leq 1$ and the coefficients $(h_j - h_{j+1})$ telescope, the sum of these coefficients is at most 1. Since the expression is a weighted sum over columns of the differences in allocation at that column, the expression is bounded by the maximum difference in any column. But because Generalized PA satisfies total variation value stability (by Lemma 22), this is bounded by $f_\ell(\lambda)$ for all subsets of advertisers, including the singleton i, as desired.

$$\begin{aligned}
|\sum_{j=1}^{k} h_j \left(P_{i,j} - P'_{i,j}\right)| &= |\sum_{j=1}^{k} \left(p_i^{(j)}(\hat{v}) - p_i^{(j-1)}(\hat{v}')\right) \left(h_j - h_{j+1}\right)| \\
&= |\max_j \left(p_i^{(j)}(\hat{v}) - p_i^{(j-1)}(\hat{v}')\right)| \leq f_\ell(\lambda)
\end{aligned}$$

◀

▶ **Lemma 22.** The k-unit PA mechanism with parameter $g(x) = x^\ell$ satisfies the property that, for all pairs of effective value vectors $\hat{v}, \hat{v}'$ and subsets of advertisers $S \subseteq [n]$,

$$|\sum_{s\in S} a_s(\hat{v}) - \sum_{s\in S} a_s(\hat{v}')| \leq \frac{\lambda^\ell - 1}{\lambda^\ell + 1} \leq f_\ell(\lambda).$$

Proof. Fix some pairs of effective value vectors $\hat{v}, \hat{v}'$ and a subset of advertisers $S \subseteq [n]$. Define E to be $\sum_{s\in S} a_s(\hat{v}) - \sum_{s\in S} a_s(\hat{v}')$ and assume without loss of generality that $E \geq 0$. We want to upper bound E by $f_\ell(\lambda)$.

First, we reduce the general case to that where the while loop never executes. That is, we modify the given instance so that the while loop never executes while only increasing E and decreasing λ. First, we can assume that $i \in S$ if $a_i(\hat{v}) > a_i(\hat{v}')$ and $i \notin S$ if $a_i(\hat{v}) < a_i(\hat{v}')$, since that those choices maximize E (and do not effect λ). We also assume that for all i, $\hat{v}_i \geq \hat{v}'_i$ and therefore $\lambda = \max_i\{\hat{v}_i/\hat{v}'_i\}$. If this is violated for $i \in S$, then raising $\hat{v}_i$ to $\hat{v}'_i$ cannot decrease E (it can only increase $\sum_{s\in S} a_s(\hat{v})$) and cannot increase λ. Similarly, if the assumption violated for $i \notin S$, then lowering $\hat{v}'_i$ to $\hat{v}_i$ cannot decrease E (it can only decrease $\sum_{s\in S} a_s(\hat{v}')$) and cannot increase λ.

Now, suppose there exists some $i \in S$ such that $\frac{k \cdot g(\hat{v}_i)}{\sum_{t=s}^{n} g(\hat{v}_t)} > 1$. Then we can reduce $\hat{v}_i$ so that $\frac{k \cdot g(\hat{v}_i)}{\sum_{t=s}^{n} g(\hat{v}_t)} = 1$ since this doesn't change E but potentially decreases λ. Finally, suppose there exists some $i \in S$ such that $\frac{k \cdot g(\hat{v}'_i)}{\sum_{t=s}^{n} g(\hat{v}'_t)} > 1$. Then consider lowering $\hat{v}'_i$ so that $\frac{k \cdot g(\hat{v}'_i)}{\sum_{t=s}^{n} g(\hat{v}'_t)} = 1$ and then scaling $\hat{v}''$ so that $\hat{v}'_i$ has its original value. This does not change E and potentially decreases λ. Therefore, we've successfully reduced to an instance in which the while loop never executes.

We now assume without loss of generality that the while loop never executes. The remaining argument follows closely from [4].

Define $\alpha := \sum_{s \in S} g(\hat{v}_s)$ and $\beta := \sum_{s \notin S} g(\hat{v}_s)$, and define α' and β' analogously. Note now that the while loop never executes, we have that for all i, $a_i(\hat{v}) = g(\hat{v}_i) / \sum_{i=1}^{n} g(\hat{v}_s)$, and similarly for $a_i(\hat{v}')$. Therefore we can write

$$E = \frac{\alpha}{\alpha + \beta} - \frac{\alpha'}{\alpha' + \beta'} = 1 - \frac{\beta}{\alpha + \beta} - \frac{\alpha'}{\alpha' + \beta'}.$$

Let $R_\alpha := \alpha / \alpha'$ and $R_\beta := \beta' / \beta$. Note that $R_\alpha \le g(\lambda)$ because for any $s \in S$, $\hat{v}_s / \hat{v}'_s \le \lambda$ so $g(\hat{v}_s)/g(\hat{v}'_s) \le g(\lambda)$. Similarly, $R_\beta \le g(\lambda)$. Observe also that our expression for E can be upper bounded by the case that these inequalities for R_α and R_β are tight.

$$\begin{aligned} E &\le 1 - \frac{\alpha \cdot g(\lambda)}{\alpha \cdot g(\lambda) + \beta'} - \frac{\alpha}{\alpha + \beta \cdot g(\lambda)} = \frac{\alpha\beta'(g(\lambda)^2 - 1)}{(\alpha + \beta' g(\lambda))(g(\lambda)\alpha + \beta')} \\ &= \frac{\alpha\beta'(g(\lambda)^2 - 1)}{g(\lambda)\alpha^2 + g(\lambda)\beta'^2 + \alpha\beta'(g(\lambda)^2 + 1)} \\ &\le \frac{\alpha\beta'(g(\lambda)^2 - 1)}{2g(\lambda)\alpha\beta' + \alpha\beta'(g(\lambda)^2 + 1)} = \frac{g(\lambda)^2 - 1}{2g(\lambda) + g(\lambda)^2 + 1} = \frac{g(\lambda) - 1}{g(\lambda) + 1}. \end{aligned}$$

Finally, we observe that $\frac{g(\lambda)-1}{g(\lambda)+1} \le f_\ell(\lambda)$, as desired:

$$E \le \frac{\lambda^\ell - 1}{\lambda^\ell + 1} = 1 - 2(\lambda^\ell + 1)^{-1} \le 1 - 2(\lambda^\ell + \lambda^\ell)^{-1} = 1 - \lambda^{-\ell} \le 1 - \lambda^{-2\ell} = f_\ell(\lambda). \qquad \blacktriangleleft$$

▶ **Theorem 24.** The Generalized PA mechanism with parameter ℓ achieves a $\left(\frac{n-k}{n}(n-k)^{-1/\ell} + 1/n\right)$-approximation to the optimal social welfare for any instance with n advertisers and k slots.

Proof. First, we consider the social welfare attained by the Generalized PA mechanism. Since $\beta_{k+1} = 0$ and $p_i^{(0)} = \vec{0}$,

$$\begin{aligned} SW(\textsc{Alg}) &= \sum_{i=1}^{n}\sum_{j=1}^{k} \alpha_i v_i \beta_j M_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{k} \hat{v}_i \beta_j \left[p_i^{(j)} - p_i^{(j-1)}\right] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} \hat{v}_i(\beta_j - \beta_{j+1}) p_i^{(j)} = \sum_{j=1}^{k} (\beta_j - \beta_{j+1}) \left(\sum_{i=1}^{n} \hat{v}_i p_i^{(j)}\right). \end{aligned}$$

Lemma 25 proves the approximation ratio of the k-unit PA mechanism. Observe that this ratio is decreasing in k. Therefore, for any j, $\left(\sum_{i=1}^{n} \hat{v}_i p_i^{(j)}\right)$ is at least an $\eta = \left(\frac{n-k}{n}(n-k)^{-1/\ell} + 1/n\right)$ fraction of Unfair-Opt. Therefore, we have

$$\begin{aligned}\text{SW}(\textsc{Alg}) &= \sum_{j=1}^{k}(\beta_j - \beta_{j+1})\left(\sum_{i=1}^{n} \hat{v}_i a_i^{(j)}\right) \geq \eta \sum_{j=1}^{k}(\beta_j - \beta_{j+1})\left(\hat{v}_1 + \cdots + \hat{v}_j\right) \\ &= \eta \sum_{j=1}^{k} \hat{v}_j \beta_j = \eta\, \textsc{Unfair-Opt}.\end{aligned}$$

◀

Robustness Should Not Be at Odds with Accuracy

Sadia Chowdhury[1]
EECS Department, York University, Toronto, Canada

Ruth Urner
EECS Department, York University, Toronto, Canada

Abstract

The phenomenon of adversarial examples in deep learning models has caused substantial concern over their reliability and trustworthiness: in many instances an imperceptible perturbation can falsely flip a neural network's prediction. Applied research in this area has mostly focused on developing novel adversarial attack strategies or building better defenses against such. It has repeatedly been pointed out that adversarial robustness may be in conflict with requirements for high accuracy. In this work, we take a more principled look at modeling the phenomenon of adversarial examples. We argue that deciding whether a model's label change under a small perturbation is justified, should be done in compliance with the underlying data-generating process. Through a series of formal constructions, systematically analyzing the relation between standard Bayes classifiers and robust-Bayes classifiers, we make the case for adversarial robustness as a locally adaptive measure. We propose a novel way defining such a locally adaptive robust loss, show that it has a natural empirical counterpart, and develop resulting algorithmic guidance in form of data-informed adaptive robustness radius. We prove that our adaptive robust data-augmentation maintains consistency of 1-nearest neighbor classification under deterministic labels and thereby argue that robustness should not be at odds with accuracy.

2012 ACM Subject Classification Theory of computation → Machine learning theory

Keywords and phrases Statistical Learning Theory, Bayes optimal classifier, adversarial perturbations, adaptive robust loss

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.5

Related Version *Previous Version*: https://arxiv.org/abs/2106.13326

Funding This research was supported by an NSERC discovery grant.

Acknowledgements Ruth Urner is also faculty affiliate member of Toronto's Vector Institute.

1 Introduction

Deep learning methods have enjoyed phenomenal successes on a wide range of applications of predictive tasks in the past decade. However, it has been demonstrated that, while these networks are often highly accurate at making predictions on natural data inputs, the performance can degrade drastically when inputs are slightly manipulated [32]. Flipping a few pixels in an image, a perturbation that is not perceivable by humans, can lead to misclassification by the trained network. These unexpected, and seemingly erratic behaviors of deep learning models have caused substantial concern over their reliability and trustworthiness. Particularly so, if these models are to be employed in applications where vulnerability to manipulations may have fatal consequences (for example if learning based vision technologies are to be employed in self-driving cars).

[1] This research was done while Sadia Chowdhury was a graduate student at York University, Toronto, Canada.

Recent years have thus seen a surge in studies aiming to enhance robustness of deep learning [10, 18, 1]. Practical approaches often either smooth out a trained predictor [12, 28], or augment the training data with perturbations of natural inputs as a way to promote robustness [36, 41]. In adversarial training this is done as part of the optimization [20, 24]. On the other hand, studies on the theory of adversarial robustness have often focused on exploring unexpected gaps in statistical and computational complexity when learning under an adversarial loss as opposed to the standard binary classification loss [26, 40, 25]. Numerous studies, both theoretical and practical, have pointed out that increasing robustness often comes at a cost of lower predictive accuracy [15, 19, 27, 5].

Naturally, an important component of analyzing and exploring a real world phenomenon, such as adversarial perturbations, is formalizing it appropriately. In supervised machine learning, the learning objective is typically encoded in form of a loss function. In this work, we take a principled look at the common definition of adversarial loss. Both theoretical studies and practical heuristics developed in the context of promoting robustness to adversarial attacks are mostly aimed at a fixed notion of smoothness with a fixed degree of perturbations that the model should be made robust to. In contrast, we formally argue how the notion of what is an admissible adversarial perturbation should be *informed by the data*. That is, robustness requirements should be aligned with the underlying data-generating process. We show how such an alignment inherently requires a *locally adaptive notion of robustness, that is, a locally adaptive robust loss*.

More specifically, we start by analyzing carefully how the previously established trade-offs between accuracy and robustness depend on a chosen (fixed) robustness parameter and the probability mass close to the decision boundary of the true underlying data-generating process. We introduce a new notion to quantify this trade-off, the *margin rate* of the distribution. We prove that, given the margin rate of a distribution, a robustness parameter can be chosen so that the two predictors that are optimal with respect to accuracy and optimal with respect to robustness loss respectively, have similar loss values (in terms of both classification and robust loss). However, we also show that choosing the robustness parameter slightly too large, can result in those two optimal predictors be very different functions. They may assign different labels on half of the space (with probability 0.5 over the data-generating process). This means that, if the robustness parameter is chosen even slightly too large, any learning method that *converges to the best possible robust loss as training data set size increases, may converge to a predictor with classification error 0.5*!

This motivates our proposition of redefining the robustness requirement. We argue that *robustness is inherently a local property* and that learned predictors should thus satisfy a local notion of robustness that is in line with the underlying data-generating process. While such a requirement can not readily be phrased as a loss function (that operates on a pair of predictor and input/output data instance), we derive a natural empirical version of this requirement. This allows for evaluating the novel adaptive robustness requirement on datasets. Further, we show how our notion of locally adaptive robustness yields a natural way of determining the robustness radius for data-augmentation. This could be used either for data-augmentation as a preprocessig step or for advesarial training.

Finally, we prove that using this form of data-augmentation as a pre-processing step maintains consistency of 1-nearest neighbor classification on tasks without stochasticity in the labels. That is, a nearest neighbor classifier on an adaptively augmented dataset converges to the optimal classification accuracy, while also satisfying the requirements of the adaptive robust loss. This formally shows how our novel framework resolves the conflicts with accuracy that are inherent in any non-adaptive notions of robustness.

1.1 Overview and summary of main contributions

We introduce our formal setup, notation for loss functions, optimal predictors and notions of statistical consistency in Section 3. In Section 4, we start with a few simple constructions, exploring how robustness (and potential divergence of 0/1-optimal and robust-optimal classifiers) relates to margins and separability of the underlying data-generating distribution. Our main contributions are presented in Sections 5 and 6 and can be summarized as follows:

Margin rate and margin canonical Bayes predictor. In Section 5.1, we *introduce the notion of a margin canonical Bayes predictor* and *the margin-rate* (Definition 4). The margin canonical Bayes predictor is a classifier that is optimal both in terms of accuracy and in terms of margins (in a precise sense that we define in this section). The margin rate can be viewed as a *relaxed measure of distributional class separateness*. It is relaxed in the sense that is does not enforce a hard margin between different classes (which is an unrealistic requirement) and instead even allows for overlap between the two class-conditional marginals (resulting in stochastic labels). We then relate the margin rate to suitable choices of r. We prove that, given the margin rate, we can choose the robustness parameter so that optimal predictors for the binary loss are also close to optimal with respect to the robust loss and vice versa (Theorem 5). Further, we show that if the labels are deterministic (no overlap between the two class-conditional marginals), then these are also close *as functions*. However, we also show that the non-stochasticity of the labels is necessary for the functions to be guaranteed to be close and that choosing r slightly too large can lead to large differences in the optimal predictors (Observations 6 and 7). Subsequently, in Subsection 5.2 we argue that, if the distribution has inherently different scales of robustness in different parts of the space, then even under deterministic labels choosing r suitable according to Theorem 5 does not lead to what is intuitively desired of a robust predictor.

Redefining robustness and resolving the conflicts with accuracy. The analysis outlined above leads to our proposition to *re-define robustness as a locally adaptive requirement*. This is presented in Section 6. There, we *introduce the adaptive robust loss*, define its *empirical version*, and develop guidance for adaptive robust data augmentation. Our proposed definition implies that the optimal predictors with respect to the binary loss and the adaptive robust loss coincide. Further, we prove that our adaptive robust data-augmentation *maintains consistency* of 1-nearest neighbor classification (NN) under deterministic labels. This shows that the undesirable effect of robustness being "at odds" with accuracy is an artifact of a specific, though common, way of defining robustness. It can be avoided be letting robustness requirements be informed by the underlying data-generating process.

Illustrative visualizations. Finally, in Appendix Section A we present a set of *illustrative experiments* for the proposed data-augmentation method and adaptive robust loss in combination with training a ReLU neural network. The synthetic datasets were designed so as to highlight the occurrence of adversarial examples when the data sits on a lower dimensional manifold, a scenario that is considered one of the sources adversarial vulnerability [22]. Our experiments visually make the case for the adaptive robust loss in situations where the label classes have *different degrees of separation in different parts of the space*.

A note on generalizations. For concreteness, we focus our presentation in this work on binary classification and work with the Euclidian metric. However, our definitions and result straightforwardly generalize to multi-class classification and to other metrics (with suitably chosen covering numbers replacing the Euclidian dimension in our result on consistency under adaptive robust data-augmentation).

2 Related Work

Enhancing robustness to adversarial attacks has received an enormous amount of research attention in recent years, in particular in terms of practical advancements [10, 18, 1, 9, 21]. We will focus our discussion of prior work on studies relating to theoretical aspects of learning under a robust loss.

Numerous recent theoretical studies focus on the parametric setup and analyze how introducing a robustness requirement may affect statistical convergence of the induced loss classes [13, 29, 26, 40, 2], whereas others have focused on computational implications [4, 25]. In particular, that there can be arbitrarily large gaps between the sample complexity of learning a hypothesis with respect to classification versus roust loss [13, 26]. Several studies have derived convergence bounds for classification under adversarial manipulations for fixed hypothesis classes [16, 3, 8].

Most related to our work are recent studies that also discuss possible options (and their implications) for phrasing a robust loss [15, 19], and in particular studies that pointed out and analyzes the trade-off between accuracy and robustness [17, 33, 39]. In particular, a recent study systematically explored the relationship between (a notion of local) Lipschitzness of a nearest neighbor predictor and its robustness. Further closely related to our work are recent studies that analyze and derive properties of optimal predictors under the robust loss and their relation to nearest neighbor predictors [35, 6, 38]. The latter work studies non-parametric learning for robust classification and proposes a method of data-preprocessing, and, similar to our result for 1-Nearest Neighbor prediction, proves implied consistency. However, the pre-processing in that study consists of pruning rather than augmenting the data. However, robustness in these prior works is considered with respect to a fixed robustness parameter. In this work, we carefully argue that adversarial robustness should instead be phrased as a locally adaptive requirement. Recently, a similar argument has independently been made [7]. Ideas of a locally adaptive robustness parameter have also appeared in some practical developments on refining adversarial training [5, 14]. Our work can be viewed as providing a formal foundation to those ideas, cleanly relating the concept of adaptive robustness to the distribution that models the data-generating process, as well as formally showing how a fixed robustness parameter easily yields inconsistencies between the robust and the standard classification loss.

Finally, we note that relationship between non-parametric methods and local adaptivity is well established and our work builds on this. In particular, it has been shown shown that nearest neighbor methods' convergence can be understood and quantified in terms of local smoothness properties of the underlying data-generating process for regression [23] as well as for classification tasks [11].

3 Formal Setup

3.1 Basic notions of statistical learning

We employ a standard setup of statistical learning theory for classification. We let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the domain and $\mathcal{Y}$ (mostly $\mathcal{Y} = \{0, 1\}$) a (binary) label space. We assume that data is generated by some distribution P over $\mathcal{X} \times \mathcal{Y}$ and let $P_{\mathcal{X}}$ denote the marginal of P over $\mathcal{X}$. We let $\mathrm{supp}(P_{\mathcal{X}})$ denote the support of this marginal. Further, we use notation $\eta_P(x) = \mathbb{P}_{(x,y)\sim P}[y = 1 \mid x]$ to denote the *regression function* of P. We say that the distribution has *deterministic labels* if $\eta_P(x) \in \{0, 1\}$ for all $x \in \mathcal{X}$. A *classifier* or *hypothesis* is a function $h : \mathcal{X} \to \mathcal{Y}$. We let $\mathcal{F}$ denote the set of all Borel measurable functions from $\mathcal{X}$ to $\mathcal{Y}$ (or all functions in case of a countable domain). A *hypothesis class* is a subset of $\mathcal{F}$, often denoted by $\mathcal{H} \subseteq \mathcal{F}$.

The quality of prediction of a hypothesis on an input/output pair (x, y) is measured by a *loss function* $\ell : (\mathcal{F} \times \mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$. For classification problems, the quality of prediction is typically measured with the *binary* or *classification loss*: $\ell^{0/1}(h, x, y) = \mathbb{1}[h(x) \neq y]$, where $\mathbb{1}[\alpha]$ denotes the indicator function for predicate α.

We denote the *expected loss* (or *true loss*) of a hypothesis h with respect to the distribution P and loss function ℓ by $\mathcal{L}_P(h) = \mathbb{E}_{(x,y)\sim P}[\ell(h, x, y)]$. In particular, we will denote the true binary loss by $\mathcal{L}_P^{0/1}(h)$. The *Bayes classifier* is a (in general not unique) classifier which has the minimal true loss with regard to P. We denote the Bayes classifier with respect to the binary loss as h_P^B and it's loss, the *Bayes risk* by $\mathcal{L}_P^B = \mathcal{L}_P^{0/1}(h_P^B)$

The *empirical loss* of a hypothesis h with respect to loss function ℓ and a sample $S = ((x_1, y_1), \ldots, (x_n, y_n))$ is defined as $\mathcal{L}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(h, x_i, y_i)$.

Further, we use the following notation to denote the set of domain points on which two classifiers differ: $h\Delta h' := \{x \in \mathcal{X} \mid h(x) \neq h'(x)\}$.

A *learner* $\mathcal{A}$ is a function that maps a finite sequence of labeled instances $S = ((x_1, y_1), \ldots$ $\ldots, (x_n, y_n))$ to a hypothesis $h = \mathcal{A}(S)$. The following notion of a *consistent learner* captures a basic desirable property: as the learner sees larger and larger samples from the data-generating distribution, the loss of the learner's output should converge to the Bayes risk.

▶ **Definition 1** (Consistency). *We say that a learner $\mathcal{A}$ is* consistent *with respect to a set of distributions $\mathcal{P}$ if, for every $P \in \mathcal{P}$, every $\epsilon, \delta > 0$ we have there is a sample-size $n(P, \epsilon, \delta)$ such that, for all $n \geq n(P, \epsilon, \delta)$, we have $\mathbb{P}_{S\sim P^n}\left[\mathcal{L}_P(\mathcal{A}(S)) \leq \mathcal{L}_P^B + \epsilon\right] \geq 1 - \delta$.*

3.2 (Adversarially) robust loss

We consider the most commonly used notion of an (adversarial) robust loss [26, 37]. For a point $x \in \mathcal{X}$, we let $\mathcal{B}_r(x)$ denote the (open) ball of radius r around x. We then define the robust loss as: $\ell^r(h, x, y) = \mathbb{1}[\exists z \in \mathcal{B}_r \ : \ h(z) \neq y]$ and we let $\mathcal{L}_P^r(h)$ denote the expected robust loss of h.

As has been done in prior work, we decompose the robust loss into its error and margin components [42, 2]: We have $\ell^r(h, x, y) = 1$ if and only if h makes a mistake on x with respect to label y, or, there is an r-close instance $z \in \mathcal{B}_r(x)$ that h labels different than x, that is, x is r-close to h's decision boundary.

The first condition holds when (x, y) falls into the *error region*, $\mathrm{err}[h] = \{(x, y) \in X \times Y) \mid h(x) \neq y\}$. The second condition holds when x lies in the *margin area* of h. We define the *margin area* of h, as the subset $\mathrm{mar}[h, r] \subseteq X$ defined by

$$\mathrm{mar}[h, r] = \{x \in \mathcal{X} \mid \exists z \in \mathcal{B}_r(x) : h(x) \neq h(z)\}$$

We can define notions of a Bayes classifier, and consistency of a learner $\mathcal{A}$ with respect to the robust loss analogously to these notions for the binary loss. We will denote the robust-Bayes classifier by h_P^{rB} and the robust-Bayes risk by $\mathcal{L}_P^{rB} = \mathcal{L}_P^r(h_P^{rB})$. We will often simply refer to the Bayes predictors as the 0/1-optimal or the r-robust optimal predictors. We note that these optimal predictors are not unique, in particular in the case that the support of the marginal $P_{\mathcal{X}}$ does not cover the full space. For example, if the data-generating distribution is supported on a lower dimensional manifold, then a 0/1-optimal predictor is only uniquely determined on that manifold (and even there only with exception of 0-mass subsets and not in areas with $\eta_P(x) = 0.5$). Similarly, r-robust optimality can be fulfilled by various predictors if the data-generating distribution is strongly separable (see Definition 4). Explicit forms (analogous to the 0/1-Bayes being a threshold of the regression function) of the r-robust optimal predictor have been derived in the literature ([38]).

4 Robustness and Margins

In this section, as a warm-up, we investigate implications of the existence of a low robust-loss classifier and differences between low binary and low robust loss. We show that the optimal classifiers with respect to these losses can differ significantly, implying that optimizing for one can strongly hurt performance with respect to the other. We then analyze the relationship between the existence of robust classifiers and margin (or separability) properties of the underlying data-generating process. We argue that, while separability implies the existence of robust classifiers with respect to *some* robustness parameter r, using a fixed robustness parameter can contravene the intention of deriving predictors that are both accurate and as robust as possible.

4.1 Binary optimal versus robust optimal

It has been shown before that the definition of the r-robust loss implies that, even in situations where the 0/1-Bayes risk is 0, that is where the labels are deterministic, no classifier may have 0 robust loss [15, 33, 42, 19]: The existence of a classifier h with $\mathcal{L}_P^r(h) = 0$ implies that the distribution is *separable*, that is, $P_{\mathcal{X}}$ is supported on r-separated regions of $\mathcal{X}$ and these regions are label-homogeneous. Namely, $\mathcal{L}_P^r(h) = 0$ implies $\mathcal{L}_P^{0/1}(h) = 0$, which means that the labeling of P is deterministic. In addition, we must have $P(\mathrm{mar}[h, r]) = 0$, which implies that any point x in the support of $P_{\mathcal{X}}$ with $h(x) = 1$ has distance at least $2r$ from any point in that support with $h(x) = 0$. In this case, this function $h = h_P^B = h_P^{rB}$ is optimal with respect to both losses.

In this subsection we inspect the potential tension between robustness and accuracy with an emphasis on the role that stochasticity of the labels play in this phenomenon. We start by observing that even if the labels are not necessarily deterministic, the optimal robust loss is strictly larger than the optimal 0/1-loss if and only if a Bayes classifier does not have a strict margin.

▶ **Theorem 2.** *We have $\mathcal{L}_P^{rB} = \mathcal{L}_P^B$ if and only if there exists a 0/1-optimal classifier h_P^B with $P_{\mathcal{X}}(\mathrm{mar}[h_P^B, r]) = 0$.*

Proof. We first assume that $P_{\mathcal{X}}(\mathrm{mar}[h, r]) > 0$ for all classifiers h that are 0/1-optimal. We fix one of them and denote it by h_P^B. Then $\mathcal{L}_P^r(h_P^B) > \mathcal{L}_P(h_P^B) = \mathcal{L}_P^B$, since on every point in its margin area, h_P^B suffers binary loss at most 0.5, while it suffers robust loss 1. Outside the margin area the loss contributions are identical for both loss functions. Furthermore, for any classifier h that is not 0/1-optimal, we have $\mathcal{L}_P^r(h) \geq \mathcal{L}_P^{0/1}(h) > \mathcal{L}_P^B$. Thus, independently of whether an optimal robust classifier h_P^{rB} is also 0/1-optimal or not, we have $\mathcal{L}_P^{rB} = \mathcal{L}_P^r(h_P^{rB}) > \mathcal{L}_P^B$.

As for the other direction, if there is a 0/1-optimal classifier h_P^B with $P_{\mathcal{X}}(\mathrm{mar}[h_P^B, r]) = 0$, then it follows immediately, that this classifier is also optimal with respect to the robust loss and its robust loss is identical to its binary loss. Thus $\mathcal{L}_P^{rB} = \mathcal{L}_P^B$. ◀

Moreover, we will now see, that if the data-generating distribution does not have a margin in the above strong sense, then the optimal classifiers with respect to 0/1-loss and r-robust loss can differ significantly *as functions*. The construction for the below result has (in very similar form) appeared in earlier work [42].

▶ **Theorem 3.** *Let $r > 0$ be a robustness parameter. There exist distributions P such that any predictors h_P^B and h_P^{rB} that are optimal with respect to 0/1-loss and r-robust loss respectively, satisfy $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] = \frac{1}{2}$, where $h_P^B \,\Delta\, h_P^{rB} = \{x \in \mathcal{X} \mid h_P^B(x) \neq h_P^{rB}(x)\}$ is the set of domain points on which the two optimal classifiers differ.*

Proof. We consider a distribution P, where $P_{\mathcal{X}}$ is supported (uniformly) on just two points x_0 and x_1 at distance less than r from each other. x_0 is always generated with label 0 and x_1 is always generated with label 1. Clearly, the 0/1-optimal classifier h_P^B labels accordingly: $h_P^B(x_0) = 0$ and $h_P^B(x_1) = 1$, resulting in $\mathcal{L}_P^{0/1}(h_P^B) = 0$. However, this classifier has largest possible r-robust loss: $\mathcal{L}_P^r(h_P^B) = 1$, since both points are at distance less than r from a point that h_P^B labels differently. On the other hand, any constant function h_c has robust loss $\mathcal{L}_P^r(h_c) = 1/2$, since it's margin has weight 0 and it mislabels with probability $1/2$. This is optimal with respect to the r-robust loss. Thus, we showed that $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] = \frac{1}{2}$. ◀

The example in the above proof shows that binary and robust optimal predictors can differ in half the area of the space. In particular, when the robustness radius r is not chosen suitably, optimizing for one can be strongly sub-optimal (incurring regret of $1/2$) for the other. This means that *any learning method, will be inconsistent with respect to at least one of the two losses in question.*

Of course, in the above example, the robustness parameter and distribution are constructed to not match suitably.

5 Relaxations of separability and the margin canonical Bayes

Strict separability between the label classes, as considered in the previous section, is a very strong assumption. We extend and refine the arguments in the previous section by relaxing this requirement and showing that, one can choose the robustness parameter r in dependence on "how separable" (in a precise sense that we introduce next) the distribution P is and on how close we would like the optimal predictors to be.

5.1 Choosing a robustness parameter

Note that, for a fixed predictor h, we have $P_{\mathcal{X}}(\mathrm{mar}[h, r]) \geq P_{\mathcal{X}}(\mathrm{mar}[h, r'])$ if $r \geq r'$. Thus, we can define a function

$$\phi_P^h(r) = P_{\mathcal{X}}(\mathrm{mar}[h, r])$$

which will monotonically decrease to 0 as r goes to 0 for any predictor h. If h is a Bayes predictor, then the rate at which $\phi_P^h(r)$ converges to 0 as $r \to 0$, can be viewed as a measure of "how separable" the data- generating process is, that is, how fast the density of the marginal $P_{\mathcal{X}}$ vanishes towards the boundary between the two label classes. However, since the Bayes predictor is generally not uniquely defined, we need to specify which Bayes predictor should be employed to serve as a measure of the separability of the distribution. For simplicity, we will assume here that we have $\eta_P(x) \neq 0.5$ for the regression function with probability 1. Then we define a margin-canonical Bayes predictor as follows: We let $\mathcal{X}^0 \subseteq \mathrm{supp}(P_{\mathcal{X}})$ denote the closure of the part of the space, where $\eta_P(x) < 0.5$ and let $\mathcal{X}^1 \subseteq \mathrm{supp}(P_{\mathcal{X}})$ the closure of the part of the space where $\eta_P(x) > 0.5$. That is, under the above assumption, the support of the marginal $P_{\mathcal{X}}$ is $\mathcal{X}^0 \cup \mathcal{X}^1$.

We can now define a *margin-canonical Bayes classifier* h_P^B by nearest neighbor labeling with respect to the sets $\mathcal{X}^0$ and $\mathcal{X}^1$. We only need to specify $h_P^B(x)$ for points x that are outside the support of $P_{\mathcal{X}}$. By definition, there exists a ball of some radius r around such a point x that has has no probability mass: $P_{\mathcal{X}}(\mathcal{B}_r(x)) = 0$. Thus, x has positive distance to both $\mathcal{X}^0$ and $\mathcal{X}^1$ and we will set $h_P^B(x) = i$ if $\mathcal{X}^i$ is the closer set to x, breaking ties arbitrarily. We note that our definitions and results in subsequent sections also hold for the margin rate of any other Bayes classifier.

▶ **Definition 4** (Margin rate). *Let P be a distribution over $\mathcal{X} \times \{0,1\}$ and let h_P^B be the margin-canonical Bayes classifier. Then we define* margin-rate *of P as the function $\Phi_P(r) = \phi_P^{h_P^B}(r)$. If there exists an $r > 0$ such that $\Phi_P(r) = 0$, we call the distribution P* strongly separable.

The margin rate is related the notion of *Probabilistic Lipschitzness* [34] and the *geometric noise exponent* [31]. We now show how the margin rate can be used for the suitable choice of robustness parameter r. We show below how to choose a robustness parameter for which the optimal robust predictor has close to optimal classification loss and vice versa. If the labels of the distribution are deterministic, then we also get closeness as functions of the optimal predictors.

▶ **Theorem 5.** *Let P be a data-generating distribution over $\mathcal{X} \times \{0,1\}$, let $\Phi_P : \mathbb{R}^+ \to [0,1]$ denote its margin rate, and let h_P^B denote the 0/1-optimal classifier defining the margin rate. For every $\epsilon > 0$, if we let $r \in \Phi_P^{-1}([0,\epsilon])$, then for any r-robust optimal classifier h_P^{rB} we have $\mathcal{L}_P^r(h_P^B) \leq \mathcal{L}_P^{rB} + \epsilon$ and $\mathcal{L}_P^{0/1}(h_P^{rB}) \leq \mathcal{L}_P^B + \epsilon$.*
In addition, if the labeling of P is deterministic, we have $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] \leq \epsilon$.

Proof of Theorem 5. Due to the way we chose the robustness parameter r here, we immediately get

$$\mathcal{L}_P^r(h_P^B) \leq \mathcal{L}_P^{0/1}(h_P^B) + \epsilon = \mathcal{L}_P^B + \epsilon$$

since $P(\mathrm{mar}[h_P^B, r]) \leq \epsilon$. We need to argue, that no other classifier h can have significantly smaller robust loss. As in the proof of Theorem 2, we observe that, we have $\mathcal{L}_P^r(h) \geq \mathcal{L}_P^{0/1}(h) \geq \mathcal{L}_P^B$ for any classifier h. Thus, in particular $\mathcal{L}_P^r(h_P^{rB}) = \mathcal{L}_P^{rB} \geq \mathcal{L}_P^B$, which yields the first claim.

For the second inequality observe that h_P^B has r-robust loss at most $\mathcal{L}_P^B + \epsilon$ by choice of r. Any robust-optimal classifier h_P^{rB} therefore has robust loss at most $\mathcal{L}_P^B + \epsilon$, which implies that its binary loss is bounded by the same quantity.

Now we assume that the labeling of P is deterministic. This implies that $\mathcal{L}_P^{0/1}(h_P^B) = 0$, thus $\mathcal{L}_P^r(h_P^B) = \mathcal{P}_{\mathcal{X}}(\mathrm{mar}[h_P^B, r])$. Let h_P^{rB} be a robust-optimal classifier. By definition of being robust-optimal, we have $\mathcal{L}_P^r(h_P^{rB}) \leq \mathcal{L}_P^r(h_P^B) = \mathcal{P}_{\mathcal{X}}(\mathrm{mar}[h_P^B, r]) \leq \epsilon$. Thus, in particular $\mathcal{L}_P^{0/1}(h_P^{rB}) \leq \epsilon$, which, in the case of deterministic labels implies $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] \leq \epsilon$. ◀

We next argue that, while a separability assumption can yield closeness in loss values of the optimal predictors, it implies closeness of the actual functions only if the labeling is also deterministic. That is, the assumption of deterministic labels is *necessary* for the second part of the above Theorem (Observation 6). More specifically, the result in the observation below shows that, a non-adaptive robustness parameter that will guarantee closeness of functions as in the first part of the above theorem, can not be determined as a function of the marginal distribution, but depends on a combination of the marginal and the "noise rate".

▶ **Observation 6.** *Let $\epsilon > 0$ be given. Then, for any γ with $0 < \gamma < \epsilon$, there exists a data-generating distribution P over $\mathbb{R}^2 \times \{0,1\}$ with linear margin rate $\Phi_P : \mathbb{R}^+ \to [0,1]$, $\Phi_P(r) = \min\{r, 1\}$ such that, for any $r \in \Phi_P^{-1}((\gamma, \epsilon))$, we get $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] = \frac{1}{2}$*

Proof. We consider a uniform marginal over two rectangles in $\mathbb{R}^2$: We set $R_1 = [-2,-1] \times [-1,1]$ and $R_2 = [1,2] \times [-1,1]$. Further, we set the regression function

$$\eta(x_1, x_2) = \begin{cases} \frac{1}{2} + \gamma \text{ if } x_2 \geq 0 \\ \frac{1}{2} - \gamma \text{ if } x_2 \leq 0. \end{cases}$$

Now it follows that a 0/1-optima predictor is $h_P^B = \mathbb{1}[x_2 \geq 0]$ while, for any $r > \gamma$, we have $h_P^{rB} = \mathbb{1}[x_1 \geq 0]$, thus $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] = \frac{1}{2}$. ◀

Next, we argue that, even under deterministic labels, choosing a robustness parameter slightly larger than implied by Theorem 5, can yield largely differing optimal predictors. The same construction as in the proof of Theorem 3 shows the following statement:

▶ **Observation 7.** *Let $\epsilon > 0$ be given. There exists a distribution P over $\mathbb{R} \times \{0,1\}$ that is strongly separable, such that, for any $r > \sup \Phi_P^{-1}([0,\epsilon])$, we have $P_{\mathcal{X}}[h_P^B \,\Delta\, h_P^{rB}] = \frac{1}{2}$.*

5.2 Towards local robustness

We now argue that, even if the distribution is strongly separable and the labels are deterministic, then choosing a uniform robustness parameter may not result in the desired outcomes in the following sense (see Figure 5.2): a classifier may be optimal with respect to the largest possible *fixed robustness parameter* (the orange classifier), but have a decision boundary that is unnecessarily close in some parts of the space where a larger *local robustness* would have been possible. To argue more formally, we consider a distribution over domain $\mathbb{R}^2 \times \{0,1\}$, where the support is distributed uniformly on four points, $((-1,0.9),0),((-1,1.1),1),((1,0.9),0),((1,2),1)$. Then predictor $h(x_1,x_2) = \mathbb{1}[x_2 \geq 1]$ is $0/1$-optimal and also r-robust optimal for any $r \leq 0.1$. However, we may prefer a predictor h^* that keeps a larger distance from the point $(1,.9)$, and is equally optimal with respect to the 0.1-robust loss.

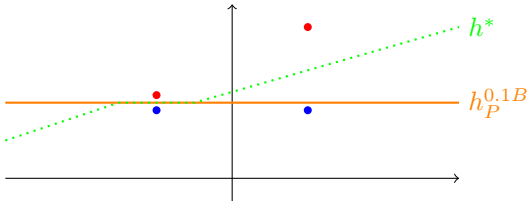

Figure 1 A robustness requirement with a uniform robustness radius is unsuitable here.

6 Redefining the Robustness Requirement

We have argued (Sections 4.1 and 5.1) that using a fixed robustness parameter r can lead to inconsistencies (in the sense that the optimal predictors with respect to binary and robust loss differ vastly) and that even under conditions where the optimal predictors can coincide (strong separability or suitably chosen robustness parameter), optimizing for the robust loss can lead to classifiers that do not reflect our intuition about an optimally robust predictor (Section 5.2). Ideally we would like a learned predictor to be *everywhere as robust as possible* (in the sense of the illustration in Figure 5.2). We will next formalize this intuition using the notions of the margin canonical Bayes and the margin rate, that we developed in the previous section.

6.1 A local robustness objective

We propose to phrase robustness in relation to a margin-canonical Bayes predictor. The core idea behind our definition is the following: If a margin-canonical Bayes predictor assigns a constant label in a ball $\mathcal{B}_r(x)$ around point x, then a robust predictor h should do the same (and only then!). For a predictor h and $x \in \mathcal{X}$, we let $\mathcal{B}^h(x)$ denote the largest ball around x on which h assigns a constant label (possibly $\mathcal{B}^h(x) = \{x\}$).

▶ **Definition 8** (Adaptive robustness). *Let P be a data-generating distribution h^B_P denote a margin-canonical Bayes predictor, and h an arbitrary predictor. We define the* adaptive robust loss ℓ^{ar} *as*

$$\ell^{ar}(h,x,y) = \mathbb{1}\left[h(x) \neq y \ \vee \ \mathcal{B}^{h^B_P}(x) \not\subseteq \mathcal{B}^h(x)\right].$$

That is, h suffers adaptive robust loss on point (x,y) if it misclassifies the point or if the point is closer to the decision boundary of h than to the decision boundary of the margin-canonical Bayes h^B_P. This definition implies that h^B_P has both minimal binary loss and optimal robust loss. We note that the above proposed loss is not technically a valid loss function, since it depends on h^B_P rather than just on h, x and y. Thus, we next propose a substitute notion of empirical adaptive robust loss.

6.2 Empirical adaptive robust loss

Let $S = ((x_1,y_1),\ldots,(x_n,y_n))$ be a labeled dataset. For a labeled domain point (x,y) we let $\rho_S(x)$ denote the distance from x to its nearest neighbor with opposite (or different in the case of more than two classes) label in S:

$$\rho_S(x,y) = \min_{i\in[n]}\{\|x_i - x\| \ \mid \ (x_i,y_i) \in S, y_i \neq y\}.$$

In the (degenerate) case that no such point in S has a label different from y (that is, all points in S have the same label), we set $\rho_S(x,y)$ to ∞ (or the diameter of the space). Note that $\rho_S(x,y)$ is well defined for points $(x,y) = (x_i,y_i) \in S$ from the dataset S itself. We now expand the dataset S by replacing each point with a (constant labeled) ball of radius $c \cdot \rho_S(x_i,y_i)$, for some (to be chosen) constant c.

▶ **Definition 9** (c-Adaptive robust expansion). *Let $S = ((x_1,y_1),\ldots,(x_n,y_n))$. We call the collection $S^c = ((\mathcal{B}_{c\cdot\rho_S(x_1,y_1)}(x_1),y_1),\ldots,(\mathcal{B}_{c\cdot\rho_S(x_n,y_n)}(x_n),y_n))$ the* c-adaptive robust expansion of S.

It is easy to see that, as long as $c \leq 1/2$, balls in the c-adaptive robust expansion of S overlap only if they have the same label. Thus, this expansion does not introduce any inconsistencies in the label requirements. Depending on the geometry of the data-generating process (eg. the curvature of the decision boundary of the regression function) we may also employ larger expansion parameters without introducing inconsistencies. Using the *c-adaptive robust expansion of S*, we can define an empirical version of the adaptive robust risk for fixed parameter c. For this, for a predictor $h : \mathcal{X} \to \mathcal{Y}$ and label y, we let $h^{-1}(y) \subseteq \mathcal{X}$ denote the part of the domain that h labels with y.

▶ **Definition 10** (Empirical c-adaptive robust loss). *Let c be an expansion parameter, $S = ((x_1,y_1),\ldots,(x_n,y_n))$ and $h : \mathcal{X} \to \mathcal{Y}$. We define the* empirical c-adaptive robust loss of h on S *as*

$$\mathcal{L}^{c-ar}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i) \not\subseteq h^{-1}(y_i)\right].$$

That is, a point $(x_i,y_i) \in S$ is counted towards the empirical c-adaptive robust empirical risk, if h does not label the whole ball $\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i)$ in the expanded set with y_i.

Remark on connection between the population and empirical version

As is usual for an empirical loss, the empirical adaptive robust loss as defined above for $c = 0.5$ corresponds to the adaptive robust loss on the empirical distribution (that is a uniform distribution of the finite data sample).

6.3 Adaptive robust data-augmentation

While the empirical c-adaptive robust risk is well defined for any predictor h and dataset S, it may, computationally, not be straightforward to verify the condition $\mathbb{1}\left[\mathcal{B}_{c\cdot\rho_S(x,y)}(x,y) \nsubseteq h^{-1}(y)\right]$. A natural estimate is to use m uniform sample points $z^1, \ldots, z^m$ from the ball $\mathcal{B}_{c\cdot\rho_S(x,y)}(x)$ and verify whether h labels all of these with y. Similarly, for training purposes, we may want to use a sample version of the c-adaptive robust expansion of S. We call this the *m-sample-c-adaptive robust augmentation of S*. The so augmented dataset S^{mc} is a set of labeled domain points and can be used as a training data-set for a standard learning algorithm.

▶ **Definition 11** (Adaptive robust data augmentation)**.** *Let $S = ((x_1, y_1), \ldots, (x_n, y_n))$ be a labeled dataset, and $m \in \mathbb{N}$. We call the collection $S^{mc} = ((z_1^1, y_1), \ldots, (z_1^m, y_1), \ldots (z_n^1, y_n), \ldots, (z_n^m, y_n))$, where every z_i^j is uniformly sampled from the ball $\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i)$, the* m-sample-c-adaptive robust augmentation of S.

To visualize the adaptive robust augmentation and its effects, we generated data from a "lower-dimensional manifold" in two dimensions, see Figure 2. It has been conjectured that the data being supported on a lower-dimensional manifold is a source of the phenomenon of vulnerability to small perturbations [22], which our visualization illustrates. The original support (the data-manifold) of data generating distributions can be seen as the green and blue lines in the first column of Figure 2, blue and green points representing points from the two classes. We trained a ReLU Neural Network with 2-hidden layers (of 10 neurons each) data points drawn from these shapes. The labeling behavior of the trained network is visualized over the ambient space in red and purple. The first column depicts the original, labeled data sets together with the networks trained on the original data. The next columns show the effect of augmentation and training with a fixed robustness parameter while the last column shows the adaptive robust augmentation.

The sequence of trained network illustrates how without augmentation, the network's decision boundary passes close to the data-manifold in several areas, yielding areas of adversarial vulnerability. The augmentation with fixed robustness, does not change this for small robustness radius. For larger, fixed robustness radius, the augmentation leads to blurring the labels. The last column shows how the adaptive robust augmentation changes the decision boundary of the trained network in the ambient space to "curve away" from the lower dimensional data manifold. Importantly the prediction on the data manifold remains unchanged. Thus the adaptive robust augmentation yields robustness without negatively affecting the accuracy of the predictor on the data-generating distribution.

We conjecture that most learners, that are consistent with respect to binary loss, remain consistent when fed a c-adaptive robust augmentation of S for $c \leq 1/2$. We prove this for a 1-nearest neighbor classification under deterministic labels. This result serves as evidence that our adaptive data augmentation does not induce any inconsistencies with the accuracy requirements. It holds for a c-robust augmentation and any m-sample-c-robust augmentation if $c \leq 0.5$. The proof has been moved to Appendix C.

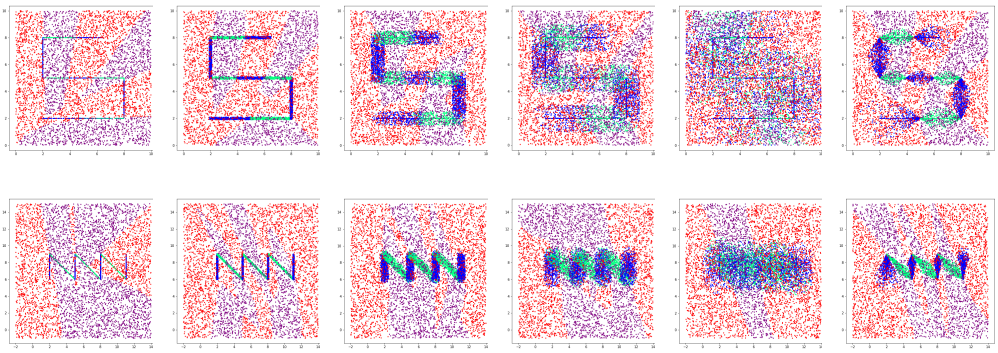

Figure 2 ReLU networks trained on data from a one-dimensional manifold, labeled with two classes (blue and green here). Left to right: original data, incrasing fixed augmentation parameters, and adaptive robust robust augmentation.

▶ **Theorem 12.** *Let P be a distribution over $[0,1]^d \times \{0,1\}$ with deterministic labels and margin rate $\Phi_P(r)$. Let $\epsilon, \delta > 0$ be given. Then, with probability at least $1-\delta$ over an is an i.i.d. sample S of size $n \geq \frac{3^d d^{0.5d}}{e\Phi_P^{-1}(\epsilon)^d \epsilon \delta}$ from P, the a 1-nearest neighbor predictor $h^{0.5}_{\mathrm{NN}}$ on a m-sample-0.5-adaptive robust augmentation of S satisfies $\mathcal{L}_P^{0/1}(h^{0.5}_{\mathrm{NN}}) \leq \epsilon$ for any $m \geq 1$.*

We will employ a similar proof technique as in Chapter 19 of [30]. In particular, we will employ Lemma 19.2 therein:

▶ **Lemma 13** (Lemma 19.2 in [30])**.** *Let $C_1, C_2, \ldots C_t$ be a collection of subsets of some domain set $\mathcal{X}$. Let D be a distribution over $\mathcal{X}$ and S be an iid sample from P of size n. Then $\mathbb{E}_{S\sim D^n}\left[\sum_{i:C_i\cap S=\emptyset} D(C_i)\right] \leq \frac{t}{n\cdot e}$.*

Recall that, for a labeled sample S, the collection $S^c = (\mathcal{B}_{c\cdot\rho_S(x_1,y_1)}(x_1,y_1),\ldots,\mathcal{B}_{c\cdot\rho_S(x_n,y_n)}(x_n,y_n))$ denotes the *c-adaptive robust expansion of S*. We will prove the theorem using this expansion for $c = 0.5$, but note, that the proof (and thus the Theorem) holds equally for

$$S^{mc} = ((z_1^1, y_1), \ldots, (z_1^m, y_1), \ldots (z_n^1, y_n), \ldots, (z_n^m, y_n)),$$

any *m-sample-c-adaptive robust augmentation of S* (where every z_i^j is uniformly sampled from the ball $\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i)$).

Proof of Theorem 12. Let P be a distribution over $[0,1]^d \times \{0,1\}$ with deterministic labels and margin rate $\Phi_P(\cdot)$. We let h_P^B be a margin optimal Bayes predictor for P. Note that, since the labels of P are deterministic $\mathcal{L}_P^{0/1}(h_P^B) = 0$. Further, we let ϵ and δ be given and set $r = \Phi_P^{-1}(\epsilon)$ (to mean the largest r, such that $\Phi_P(r) \leq \epsilon$). Further, we set $r' = r/3$.

We can now partition the space $[0,1]^d$ into $t = \left(\frac{\sqrt{d}}{r'}\right)^d$ many sub-cubes of side-length $r'/\sqrt{d}$ and thus diameter r'. We denote the cells in this partition by $C_1, \ldots, C_t$.

We now let S be a labeled sample and let $h_S^c = h_S^5$ be the nearest neighbor classifier on the .5-adaptive robust expansion of S. We now bound the mass of points x on which h_S^c makes a false classification by noting that $h_S^c(x) \neq h_P^B(x)$ implies that one of these three conditions hold:

C1: x falls into a cell C_k that has empty intersection with the sample S;
C2: there is at least one sample point $(x_i, y_i) \in S$ in the same cell C_k as x, and there exists at least one such $(x_i, y_i) \in S$ with $y_i \neq h_P^B(x)$;
C3: there is at least one sample point $(x_i, y_i) \in S$ in the same cell C_k as x, and we have $y_i = h_P^B(x)$ for all (x_i, y_i) in the same cell, but there is another sample point $(x_j, y_j) \in S$ (in a different cell) with $y_j \neq h_P^B(x)$ and x is closer to the expansion $\mathcal{B}_{c\cdot\rho_S(x_j,y_j)}(x_j, y_j)$ of x_j than to the expansion $\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i, y_i)$ for all x_i in C_k.

If S is an iid sample from P, then, by Lemma 13 the expected mass of points x cells that are not hit by the sample S is bounded by $\frac{t}{n\cdot \mathrm{e}} = \frac{3^d\sqrt{d}^d}{\Phi_P^{-1}(\epsilon)^d\cdot n\cdot \mathrm{e}}$. By Markov's inequality, this implies

$$\mathbb{P}_{S\sim P^n}\left[\sum_{i:C_i\cap S=\emptyset} P_{\mathcal{X}}(C_i) > \epsilon\right] \leq \frac{3^d\sqrt{d}^d}{\epsilon\cdot\Phi_P^{-1}(\epsilon)^d\cdot n\cdot \mathrm{e}}.$$

Setting this to δ shows that, with probability at least $1-\delta$ over a sample S of size $n \geq \frac{3^d\sqrt{d}^d}{\epsilon\cdot\Phi_P^{-1}(\epsilon)^d\cdot\delta\cdot \mathrm{e}}$ the mass of points that fall into "error case" C1 is bounded by ϵ.

We now argue that the mass of points that fall into "error case" C2 or C3 is also bounded by ϵ by showing that such points actually fall into the r-margin area of h_P^B and, by choice of r and by definition of Φ_P, we have $P_{\mathcal{X}}(\mathrm{mar}[r, h_P^B]) \leq \epsilon$.

Consider a point x in case C2. If there exist a point $(x_i, y_i) \in S$ in the same cell as x with $y_i \neq h_P^B(x)$, then by the choice of the size of the cells $x \in \mathrm{mar}[r', h_P^B] \subseteq \mathrm{mar}[r, h_P^B]$.

Now consider a point x in case C3: There exists at least one point $(x_i, y_i) \in S$ in the same cell as x and all points in the same cell as x have label $h_P^B(x)$. But there is another sample point $(x_j, y_j) \in S$ (in a different cell) with $y_j \neq h_P^B(x)$ and x is closer to the expansion $\mathcal{B}_{c\cdot\rho_S(x_j,y_j)}(x_j, y_j)$ of x_j than to the expansion $\mathcal{B}_{c\cdot\rho_S(x_i,y_i)}(x_i, y_i)$ for any x_i in the same cell as x, where $c = 0.5$.

Recall that $\rho_S(x_j, y_j)$ is the distance between x_j and a closest point in S of opposite label to y_j. We now set $\rho = 0.5\cdot\rho_S(x_j, y_j)$ for short, that is ρ is the radius of the expansion of (x_j, y_j). Since the cell that x is in also contains (x_i, y_i) and $y_i \neq y_j$ in this case C3, we know that

$$2\rho \leq \|x_i - x_j\|. \tag{1}$$

Further, we know

$$\|x_i - x\| \leq r' \tag{2}$$

since x_i in in the same cell as x.

Let $z \in \mathcal{B}_{c\cdot\rho_S(x_j,y_j)}(x_j, y_j)$ be the point in $\mathcal{B}_{c\cdot\rho_S(x_j,y_j)}(x_j, y_j)$ closest to x. Note that, since z in in the expansion of x_j, we have

$$\|z - x_j\| \leq \rho. \tag{3}$$

Then, since x is closer to the expansion of x_j than the expansion of x_i, we can infer, using Equation 2 that

$$\|x - z\| \leq \|x - x_i\| \leq r' = r/3. \tag{4}$$

This implies using the triangle inequality and Equations 4 and 2 that

$$\|z - x_i\| \leq \|z - x\| + \|x - x_i\| \leq 2r'. \tag{5}$$

Now, by again using the triangle inequality and Equations 3 we get

$$\|x_i - x_j\| \leq \|x_i - z\| + \|z - x_j\| \leq \|x_i - z\| + \rho, \tag{6}$$

Thus, using Equations 1 and then Equation 6, we get

$$2\rho \leq \|x_i - x_j\| \leq \|x_i - z\| + \rho$$

which immediately implies $\rho \leq \|x_i - z\|$. Together with Equation 5 the above yields: $\rho \leq 2r'$. Now, again invoking the triangle inequality and using Equations 4 and 3, we can bound the distance between x and x_j:

$$\|x - x_j\| \leq \|x - z\| + \|z - x_j\| \leq r' + 2r' = r.$$

Thus, in this case, x also falls into the r-margin area of h_P^B since $h_P^B(x) \neq h_P^B(x_j)$. This concludes the proof of the Theorem. ◀

7 Concluding Remarks

In this work, we provide a formal foundation for adversarial robustness as an *adaptive requirement*. We argue for re-framing adversarial robustness as a requirement that should be in line with the underlying distribution's margin properties. We do this by introducing a novel notion of the *margin-rate* that quantifies probability mass in proximity to a Bayes optimal's decision boundary in a more flexible way than standard notions of margin-separability do. We employ this measure to propose a formal notion of such an adaptive loss, as well as an accompanying empirical version and implied data-augmentation paradigm. As a first sound justification of this proposal, we prove that this type of adaptive data-augmentation maintains consistency of a non-parametric method (namely 1-nearest neighbor classification under deterministic labels). We believe this to be a natural and useful take on resolving the discrepancies with accuracy that have been reported in the context of adversarial robustness (both in theoretical and practical studies). Further, we believe that our notion of a data-informed, adaptive robustness radius might be useful for other methods that employ data augmentation.

References

1 Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

2 Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Black-box certification and learning under adversarial perturbations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 2020.

3 Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In *Algorithmic Learning Theory, ALT*, pages 162–183, 2019.

4 Pranjal Awasthi, Abhratanu Dutta, and Aravindan Vijayaraghavan. On robustness to adversarial examples and polynomial optimization. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 13760–13770, 2019.

5 Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *CoRR*, abs/1910.08051, 2019. `arXiv:1910.08051`.

6 Robi Bhattacharjee and Kamalika Chaudhuri. When are non-parametric methods robust? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 2020.

7 Robi Bhattacharjee and Kamalika Chaudhuri. Consistent non-parametric methods for adaptive robustness. *CoRR*, abs/2102.09086, 2021. `arXiv:2102.09086`.

8 Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya P. Razenshteyn. Adversarial examples from computational constraints. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 831–840, 2019.

9 Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019. `arXiv:1902.06705`.

10 Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *CoRR*, abs/1810.00069, 2018. `arXiv:1810.00069`.

11 Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems, NIPS*, pages 3437–3445, 2014.

12 Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 1310–1320, 2019.

13 Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 230–241, 2018.

14 Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA training: Direct input space margin maximization through adversarial training. In *8th International Conference on Learning Representations, ICLR*, 2020.

15 Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Advances in Neural Information Processing Systems 31, NeurIPS*, pages 10359–10368, 2018.

16 Uriel Feige, Yishay Mansour, and Robert Schapire. Learning and inference in the presence of corrupted inputs. In *Conference on Learning Theory, COLT*, pages 637–657, 2015.

17 Yarin Gal and Lewis Smith. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with bayesian neural networks, 2018. `arXiv:1806.00667`.

18 Ian J. Goodfellow, Patrick D. McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, 2018.

19 Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 7444–7453, 2019.

20 Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári. Learning with a strong adversary. *CoRR*, abs/1511.03034, 2015. `arXiv:1511.03034`.

21 Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32: NeurIPS*, pages 125–136, 2019.

22 Marc Khoury and Dylan Hadfield-Menell. Adversarial training with voronoi constraints. *CoRR*, abs/1905.01019, 2019. `arXiv:1905.01019`.

23 Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems, NIPS*, pages 729–737, 2011.

24 Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR*, 2018.

25 Omar Montasser, Surbhi Goel, Ilias Diakonikolas, and Nathan Srebro. Efficiently learning adversarially robust halfspaces with noise. *arXiv preprint*, 2020. `arXiv:2005.07652`.

26 Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory, COLT*, pages 2512–2530, 2019.

27 Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 13824–13833, 2019.

28 Hadi Salman, Jerry Li, Ilya P. Razenshteyn, Pengchuan Zhang, Huan Zhang, Sébastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 11289–11300, 2019.

29 Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 5014–5026, 2018.

30 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

31 Ingo Steinwart and Clint Scovel. Fast rates for support vector machines using gaussian kernels. *The Annals of Statistics*, 35(2):575–607, 2007.

32 Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR*, 2014.

33 Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR*, 2019.

34 Ruth Urner, Sharon Wulff, and Shai Ben-David. PLAL: cluster-based active learning. In *COLT 2013 - The 26th Annual Conference on Learning Theory*, pages 376–397, 2013.

35 Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 5120–5129, 2018.

36 Huanrui Yang, Jingchi Zhang, Hsin-Pai Cheng, Wenhan Wang, Yiran Chen, and Hai Li. Bamboo: Ball-shape data augmentation against adversarial attacks from all directions. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

37 Yao-Yuan Yang, Cyrus Rashtchian, Yizhen Wang, and Kamalika Chaudhuri. Adversarial examples for non-parametric methods: Attacks, defenses and large sample limits. *CoRR*, abs/1906.03310, 2019. `arXiv:1906.03310`.

38 Yao-Yuan Yang, Cyrus Rashtchian, Yizhen Wang, and Kamalika Chaudhuri. Robustness for non-parametric classification: A generic attack and defense. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 941–951, 2020.

39 Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R. Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *Advances in Neural Information Processing Systems 33 NeurIPS*, 2020.

40 Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher complexity for adversarially robust generalization. In *Proceedings of the 36th International Conference on Machine Learning,ICML*, pages 7085–7094, 2019.

41 Hang Yu, Aishan Liu, Xianglong Liu, Gengchao Li, Ping Luo, Ran Cheng, Jichen Yang, and Chongzhi Zhang. Pda: Progressive data augmentation for general robustness of deep neural networks, 2020. `arXiv:1909.04839`.

42 Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 7472–7482, 2019.

A Visualizations

To further validate our proposed adaptive robust data augmentation method, we present a set of illustrative experiments on various synthetic datasets. To allow for visualizations, we generate data from a "lower-dimensional manifold" in two dimensions. It has been conjectured that the data being supported on a lower-dimensional manifold is a source of the phenomenon of vulnerability to small perturbations [22]. Our visualizations in in Figure 3 illustrate this phenomenon.

The original support (the data-manifold) of the data generating distributions is one-dimensional hehre and can be seen as the green and blue lines in the first column of images in Figure 3. Blue and green points represent points from the two classes. We term our synthetic shapes in Figure 3 **Sines, S-figure, NNN, circles, boxes**. We train a ReLU Neural Network with 2-hidden layers (of 10 neurons each) data points drawn from these shapes. The labeling behavior of the trained network is visualized over the ambient space in red and purple. The first image in each row depicts the original, labeled data together with the network trained on the original data.

We see in those left-most illustrations that without any augmentation, the network's decision boundary is often located close to the data-manifold. Since the data is supported only on the lower-dimensional manifold, there is no incentive for the decision boundary to keep a distance from the data-manifold. While the network labels areas on the manifold itself correctly, this behavior leads to the existence of points that are vulnerable to adversarial perturbations: a small deviation away from the data-manifold can lead to a different labeling by the network.

We then augment the training datasets with both fixed and adaptive expansion parameter and train ReLU Neural Networks of the same size on the augmented datasets. The remaining images in each row again illustrate the augmented datasets (green and blue) together with the labeling behaviors of the resulting networks (red and purple). The last image in each row corresponds to the adaptive augmented data, while the intermediate images correspond to augmentations with increasing, but fixed expansion parameters.

For non-adaptive expansion parameter, we iteratively increase the parameter in a fixed sequence, $(0.1, 0.5, 1, 2, ..., 16)$. These expansion parameters were chosen based on the range of the attribute values in the datasets. For each sample in a d-dimensional dataset, a d-dimensional sphere is generated where the radius is the fixed-parameter and the current sample is the center of the sphere. Four new points are then generated in this sphere for each sample point. Hence, the training dataset is expanded to five times its original size after fixed-parameter expansion.

Analogously we augment the data with an adaptive expansion parameter. The key difference is in the calculation of the radius of the sphere. A fraction of the distance between the current sample and a nearest neighbor of a different class is used as the radius for the sphere generation. Each of the middle columns in Figure 3 corresponds to augmentation with a fixed expansion parameter, while the last column shows the 2/3-adaptive robust augmentation of the training data. The original training dataset contains 1000 training points and the augmented datasets 5000 data points each.

For the various networks we evaluate binary loss and the adaptive robust loss. To estimate the adaptive robust loss at a point x, we determine its distance ρ to a point in the dataset with a different label and then generate 10 test points uniformly at random from a ball

of radius $\rho/2$. If one of these gets a different label than x by the network (or if the point is mislabeled itself) it suffers adaptive robust loss 1. Table 1 summarizes the binary and adaptive robust losses of the various networks. We see that the adaptive augmentation leads consistently to the lowest binary (always rank 1) and low adaptive robust loss (rank 1 and once rank 2). This shows that the adaptive augmentation not only is not in conflict with accuracy, but empirically improves accuracy of a trained network.

Finally, we also trained ReLU neural networks on several real-world data sets from the UCI repository. For each dataset, we normalized the features to take values in [0,1]. As in the experiments on the synthetic data, we trained the networks on the original data, as well as various augmented datasets, including using the 2/3-adaptive augmentation. The datasets were split into training and test data with a ratio of $80 - 20$ respectively. In Tables 1 and 2, we report the binary and robust losses of these networks. We observe, again, that the robust

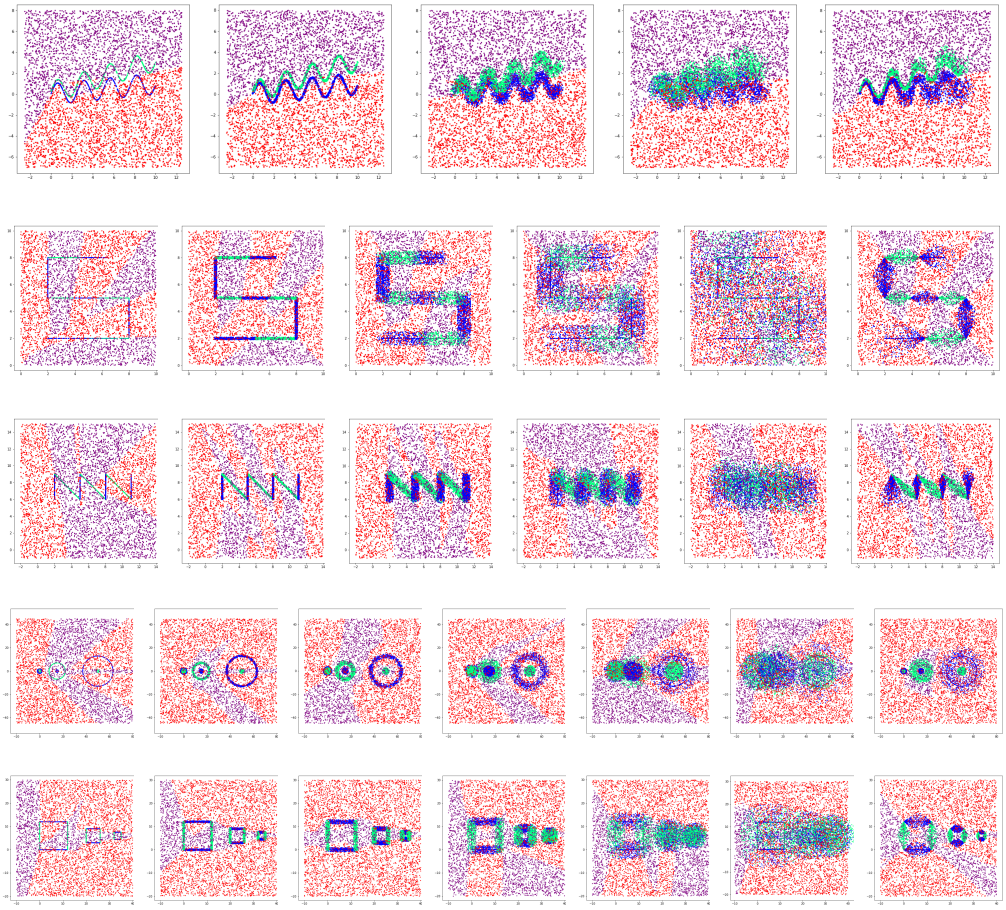

Figure 3 ReLU networks trained on data from a one-dimensional manifold in two-dimensional space, labeled using two classes (blue and green here). The various shapes by row: **Sines, S-figure, NNN, circles, boxes**. Left-most: original training data; various middle images: training data augmented using increasing expansion parameters; right-most: training data robust-adaptive expanded. We use data generated uniformly at random from the ambient space to illustrate the network's labeling (red and purple). Using just original training data, or only slightly augmented data, we observe that the network's decision boundary is often close to the manifold.

augmentation promotes the best performance in terms of 0/1 accuracy. Additionally, the adaptive robust loss is close to the best adaptive robust loss achieved with a fixed expansion parameter on each dataset. Using the adaptive augmentation can thus serve to save needing to search for an optimal expansion parameter on different tasks.

In summary, our initial experimental explorations here showed that the adaptive augmentation consistently yielded a robust predictor with best 0/1-loss. This confirms the intended design of an adaptive robustness and data augmentation paradigm that avoids the undesirable tradeoffs between robustness and accuracy.

Table 1 Overview on the binary and adaptive robust losses of the networks trained on trained on the various synthetic datasets with various augmentations.

Dataset	Expansion Radius for Training	Adaptive Robust Loss	Binary Loss
Sines	Original	0.2882	0.104
	0.1	0.1693	0.071
	0.5	0.2443	0.147
	1	0.3116	0.177
	2	0.3521	0.208
	Adaptive	**0.1403**	**0.038**
S-figure	Original	0.3516	0.044
	0.1	0.1514	0.016
	0.5	0.0429	0.027
	1	0.0844	0.05
	2	0.2373	0.21
	Adaptive	**0.0393**	**0.017**
NNN	Original	0.3841	0.2124
	0.1	0.2609	0.1086
	0.5	0.2008	0.1048
	1	0.1969	0.0952
	2	0.386	0.3714
	Adaptive	**0.08972**	**0.04**
circles	Original	0.4483	0.0133
	0.5	0.2629	**0**
	1	0.3472	0.0108
	2	**0.1778**	0.0242
	4	0.3076	0.0783
	8	0.3557	0.1733
	16	0.3054	0.1633
	Adaptive	0.254	**0**
boxes	Original	0.3427	0.08
	0.5	0.2623	0.0775
	1	0.2229	0.0775
	2	0.2252	0.1667
	4	0.2839	0.2283
	8	0.4274	0.3458
	Adaptive	**0.2077**	**0.075**

Table 2 Overview on the binary and adaptive robust losses of the networks trained on trained on the various UCI datasets with various augmentations.

Dataset	Expansion Radius for Training	Adaptive Robust Loss	Binary Loss
Iris	Original	0.0957	0.0435
	0.1	**0.0783**	**0**
	0.5	0.1304	**0**
	1	0.3478	0.087
	2	0.391	0.3478
	Adaptive	0.087	**0**
Breast Cancer	Original	0.1351	0.0263
	0.1	0.0956	**0.0175**
	0.5	0.0842	0.0351
	1	0.0833	0.0439
	2	**0.0693**	**0.0175**
	Adaptive	0.0719	**0.0175**
Bank Note Authentication	Original	0.0804	**0**
	0.1	0.0479	**0**
	0.5	0.1593	0.0909
	1	0.1153	0.0036
	2	0.1058	0.0036
	Adaptive	**0.0167**	**0**
Heart Disease	Original	**0.3465**	0.1628
	0.1	0.3791	0.2093
	0.5	0.386	0.2093
	1	0.4489	0.2791
	2	0.507	0.3488
	Adaptive	0.3604	**0.1395**
Immunotherapy	Original	0.263	0.1852
	0.1	0.2926	0.1111
	0.5	0.3482	0.1852
	1	0.2333	0.1852
	2	0.437	0.2593
	Adaptive	**0.174**	**0.0741**
Parkinsons	Original	**0.1423**	0.078
	0.1	0.1678	0.0847
	0.5	0.1542	0.0678
	1	0.2322	0.1017
	2	0.2322	0.1186
	Adaptive	0.1627	**0.0508**

Improved Generalization Guarantees in Restricted Data Models

Elbert Du[1]
Department of Computer Science, Harvard University, Boston, MA, USA

Cynthia Dwork[1]
Department of Computer Science, Harvard University, Boston, MA, USA

Abstract

Differential privacy is known to protect against threats to validity incurred due to adaptive, or exploratory, data analysis – even when the analyst adversarially searches for a statistical estimate that diverges from the true value of the quantity of interest on the underlying population. The cost of this protection is the accuracy loss incurred by differential privacy. In this work, inspired by standard models in the genomics literature, we consider data models in which individuals are represented by a sequence of attributes with the property that where distant attributes are only weakly correlated. We show that, under this assumption, it is possible to "re-use" privacy budget on different portions of the data, significantly improving accuracy without increasing the risk of overfitting.

2012 ACM Subject Classification Theory of computation → Machine learning theory; Theory of computation → Design and analysis of algorithms; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Differential Privacy, Adaptive Data Analysis, Transfer Theorem

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.6

Acknowledgements The authors are indebted to Guy Rothblum and Pragya Sur for many helpful conversations.

1 Introduction

It has been known for nearly a decade that interacting with data in a differentially private fashion provides a universal approach to reducing the risk of spurious scientific discoveries incurred by *adaptive*, or *exploratory*, data analysis [5, 6], in which new analyses or questions posed of the data depend on the outcomes of previous analyses. Strengthenings of these initial results, and extensions to other information-restrictive interactions, rapidly followed, for example, [1, 4]. In these works and their *sequelae*, the data analyst is viewed as an *accuracy adversary* whose goal is to find a query on which the dataset (or the response produced by a mechanism that interacts with the data) is not representative of the population.

For some kinds of data and analyses, for example, in Genome-Wide Association Studies (GWAS), which involve vast numbers of statistical queries on very high dimensional data, differential privacy faces daunting lower bounds [3]. However, our interest in this work is in accuracy, and not privacy *per se*. Inspired by two natural examples, we consider the question of whether we can improve on the accuracy by exploiting independence properties in the features of the data. In data streams, it is often assumed that elements far apart in the stream are uncorrelated or only weakly correlated, with the correlation decreasing as the distance increases. In a stream, data of different individuals are interleaved; genomic

[1] Corresponding Author

information has this same low-correlation property even in the DNA for a single individual: for example, chromosomes are considered to be unrelated, and even within a chromosome correlations decrease with distance [12].

While genomic data is our motivating example, we note that similar assumptions are reasonable in other settings. For example, in certain kinds of image data distant pixels may be relatively uncorrelated even within a single image. We will make this notion precise in Section 2.

The line of work described above gave rise to a number of so-called "transfer theorems," and we will make use of the sharp recent addition to this literature in [10]. Transfer theorems generally say that if a query-response mechanism satisfies some specific quantifiable constraint on the information it imparts, then an analyst interacting with this mechanism cannot overfit to within some related quantity. In the context of differential privacy the requirement is that the mechanism must be (ε, δ)-differentially private and (α', β')-sample accurate[2], and the guarantee from the theorem is that the responses will be $(\alpha = \alpha(\epsilon, \delta, \alpha'), \beta = \beta(\epsilon, \delta, \beta'))$-distributionally accurate, meaning that with probability at least $1 - \beta$ the responses are within α of their distributional values.

Our restriction on data models comes into play here: consider a genome-wide association study (GWAS), in which the dataset contains, for each of n individuals, a string of potentially millions of Single Nucleotide Polynomorphisms (SNPs). A typical study will make huge numbers of counting queries, looking for SNPs that are associated with a disease, at a huge cost in accuracy, as the data of each individual simultaneously affect all these counts. We asked the following question: under the assumption that distant SNPs in the genome of any given individual are at best very loosely correlated, is it possible to "re-use" privacy budget when examining distant portions of the genome? We will not achieve privacy in so doing, but can we achieve better accuracy? For example, if we examine the dataset one chromosome at a time, meaning, we analyze the first chromosome for everyone in the dataset using $(\varepsilon_0, \delta_0)$-DP and a single application of a transfer theorem to ensure validity on the queries for this chromosome, and then examine the second chromosome for everyone in the dataset, "re-using" $(\varepsilon_0, \delta_0)$-DP, and it really is the case that one's first and second chromosomes are unrelated, can we safely apply the transfer theorem a second time to conclude that the queries on the second chromosome have not overfit, and so on? We obtain an affirmative answer to this and other, less restrictive, data access models. The key factors in the analysis are (1) the independence of the features (chromosomes, distant SNPs) and (2) the exclusion of queries that simultaneously operate on distant features (sums of adjacent features permitted, sums of distant features not supported).

Our first result considers the model in which each individual's data is partitioned into a sequence of m fully independent blocks. Roughly speaking, it says that the privacy budget for a single block can be re-used, risking only a factor of m increase in failure probability.

▶ **Theorem 1** (Informal). *If the data consists of m independent blocks, and our mechanism M performs an (ϵ, δ)-DP and (α, β)-sample accurate interaction on each block, then M is $(\alpha', m\beta')$-distributionally accurate, where α' and β' are the parameters we get from the transfer theorem on each block.*

To build intuition for this result, suppose that, for each individual, we have a series of $m > 1$ mutually independent blocks of features $B_1, B_2, ...B_m$. That is, there are m distributions $D_1, \ldots, D_m$ and the data of each individual is a draw from the product distribution $D_1 \times D_2 \times \cdots \times D_m$. Suppose, for this intuition-building only, that the mechanism accesses the

[2] That is, with probability at least $1 - \beta'$ the responses produced are within α' of their sample values.

data in m epochs, first accessing block B_1 of attributes for all n individuals, then accessing block B_2 of attributes for all n individuals, and so on. At epoch $i \in [m]$ the analyst may carry out any (ϵ_0, δ_0)-DP analysis of the data on block i. In this case, we claim we can apply the transfer theorem m times while retaining the accuracy guarantees and paying a factor of m in the failure probability β. To see this, note that, because of the independence assumptions, we can assume that the data for block B_i have not even been selected before processing of this block. In this case, an accuracy adversary – even one with all the data of blocks $B_1, \ldots, B_{i-1}$ "hard-wired" in, is just an arbitrary adversary. Allowing this adversary to interact with an independently randomly chosen block B_i is precisely what happens in differential privacy: an adversary interacts with (apparently) freshly drawn data. We can therefore apply the transfer theorem to conclude that, on this ith block, with probability at least $1 - \beta$, the responses are α-accurate. A union bound then gives the result, yielding an upper bound of $m\beta$ on the probability of failure.

While this "thick" streaming access mode is not required for our algorithms, it remains useful for building intuition when we depart from the full independence data models.

For our most general result, we consider models in which correlations between attributes a_i and a_j in the data of a single individual falls exponentially with their "distance" $|i - j|$, and we restrict the "width" of a query so that it cannot simultaneously access very distant elements. Roughly speaking, in our model distant attributes have high probability of being independent and vanishing probability of being arbitrarily dependent. We show that we can again re-use the privacy budget, paying only a small additional probability of failure due to the low-probability dependence events.

▶ **Theorem 2** (Informal). *Suppose the probability that two attributes at distance d are not independent is negligible, and suppose further that queries involve only attributes with distance at most d. Then, if our mechanism M is (ϵ, δ)-DP and (α', β')-sample accurate on every sequence of $2d + 1$ consecutive attributes, it's also $(\alpha, m\beta + negl)$-distributionally accurate where $(\alpha = \alpha(\epsilon, \delta, \alpha'), \beta = \beta(\epsilon, \delta, \beta'))$ are the parameters we get from the transfer theorem.*

2 Preliminaries

We are interested in query answering *mechanisms* that operate on datasets and produce outputs. A standard view is that the mechanism interacts with an *adversary* whose goals are unknown and who may be malicious. Both parties may employ randomness.

The interaction between a mechanism $\mathcal{M}$ and an adversary A using sample S, is a random variable denoted by $\text{Interact}(\mathcal{M}, A; S)$, where the adversary generates queries q_i and the mechanism $\mathcal{M}$ generates responses a_i, giving rise to *transcripts* of the form $(q_1, a_1, q_2, a_2, \ldots q_k, a_k)$. Later queries may be chosen as functions of the transcript prefix. We will sometimes use the shorthand $I(S)$ when $\mathcal{M}$ and A are clear from context. The set of transcripts that can be generated by the interaction between $\mathcal{M}$ and A will be denoted $\text{Interact}(\mathcal{M}, A, *)$.

In this work, individuals are represented in the dataset as a sequence of m *attributes*, or *covariates*. Doing so allows us to formalize the idea of *distance* among attributes in a dataset as the difference in the indices of the attributes.

▶ **Definition 3.** *Datasets X and X' of the same cardinality are* adjacent *if they differ on at most one element.*

▶ **Definition 4.** *A mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for any pair of adjacent datasets X, X', any adversary A, and any set of transcripts E, we have*

$$\Pr[\text{Interact}(\mathcal{M}, A, X) \in E] \leq e^{\epsilon} \cdot \Pr[\text{Interact}(\mathcal{M}, A, X') \in E] + \delta,$$

where the probability space is over the randomness of $\mathcal{M}$ and $\mathcal{A}$.

▶ **Definition 5.** *A mechanism $\mathcal{M}$ satisfies (α, β)-sample accuracy if for every data analyst A and every data distribution $\mathcal{P}$,*

$$\Pr_{X\sim\mathcal{P}^n, \text{Interact}(\mathcal{M},A,X)}\left[\max_j |q_j(S) - a_j| \geq \alpha\right] \leq \beta.$$

Similarly, $\mathcal{M}$ satisfies (α, β)-distributional accuracy if for every data analyst A and every data distribution $\mathcal{P}$,

$$\Pr_{X\sim\mathcal{P}^n, \text{Interact}(\mathcal{M},A,X)}\left[\max_j |q_j(\mathcal{P}^n) - a_j| \geq \alpha\right] \leq \beta.$$

▶ **Definition 6.** *We say that a sequence of random variables $(B_1, B_2, \ldots, B_m)$ is k-dependent if for any two subsets I and J of $\{1, 2, \ldots, m\}$ such that $\max(I) < \min(J)$ and $\min(J) - \max(I) > k$, the families of random variables $(B_i)_{i\in I}$ and $(B_j)_{j\in J}$ are independent.*

▶ **Definition 7.** *A linear query (sometimes called statistical query) is a query q such for any individual $X \in \mathcal{X}$, $q(x) \in [0, 1]$, and for any sample $S \in \mathcal{X}^n$, $q(S) = \frac{1}{n}\sum_{x\in S} q(x)$*

From time to time, we will need to focus on the queries that involve a specific collection of attributes. For this purpose, we introduce the following definition:

▶ **Definition 8.** *Let Q be a collection of queries, defined before the interaction happens. Given a mechanism $\mathcal{M}$, the transcript of the interaction restricted to Q is defined as follows:*

1. *$\mathcal{M}$ interacts with an adversary A, producing transcript Π*
2. *As a postprocessing step, we remove every query and answer (q, a) from Π such that $q \notin Q$. Let Π' denote resulting transcript.*
3. *Π' is the transcript of the interaction restricted to Q.*

Intuitively, this is just "projecting" the transcript onto Q.

2.1 Transfer Theorem

The following is Theorem 3.5 from [10].

▶ **Theorem 9.** *Suppose M is (ϵ, δ)-DP and (α, β)-sample accurate for linear queries. Then for any data distribution $\mathcal{P}$, a sample $S \sim \mathcal{P}^n$, any analyst $\mathcal{A}$, and any constants $c, d > 0$:*

$$\Pr_{S\sim\mathcal{P}^n, \Pi\sim\text{Interact}(M,A;S)}\left[\max_j |a_j - q_j(\mathcal{P})| > \alpha + (e^\epsilon - 1) + c + 2d\right] \leq \frac{\beta}{c} + \frac{\delta}{d}$$

i.e. it is (α', β')-distributionally accurate for $\alpha' = \alpha + e^\epsilon - 1 + c + 2d$ and $\beta' = \frac{\beta}{c} + \frac{\delta}{d}$.

There are two facts to note here. Firstly, the transfer theorem assumes that all queries are linear queries (often called statistical queries in the literature). A linear query q is one in which for each $x \in S$, $q(x) \in [0, 1]$ and $q(S) = \frac{1}{n}\sum_{x\in S} q(x)$.

The notable features of a linear query are that q must be a function of x, so it is deterministic and also cannot use information not captured in the features of the database, such as index. Linear queries are powerful; it is known that we can learn nearly everything that is PAC-learnable in the statistical queries learning model [11]. In addition, there is a vast literature on handling very large numbers of differentially private statistical queries, beginning with the exciting contributions in [2, 8].

Note that, were we to remove the constraint that the query must be a function only of the covariates (and not, say the index of a row in the database), the sample accuracy of the mechanism would become ill-defined.

The other key fact is that, in the statement of the transfer theorem, the probability is taken over both the sample and the randomness employed during the interaction. Thus, the mechanism could be arbitrarily bad for some particularly unrepresentative sample. That is, we could come up with "counterexample" samples where we do get $a_j - q_j(\mathcal{P})$ to be very large (imagine a sample where $\alpha + \alpha'$ is significantly greater than $|q(S) - q(\mathcal{P})|$ for many queries q).

In the following sections, we will analyze mechanisms, where, to bound their privacy loss naïvely, we would need to take the composition of m mechanisms, requiring us to pay an $\Omega(\sqrt{m})$ factor in the DP guarantee. By assuming (limited) independence in our data, we are able to instead bound the privacy loss with the composition of 1 or 2 mechanisms, while having the same m-fold increase in the probability of failure that we would get from composition.

3 Full Independence

In this setting, we are motivated by the structure of chromosomes. The entire sequence of DNA is contained in many linear chromosomes, and there is no known dependence between the sequence of one linear chromosome and the sequences of any other linear chromosomes. As such, it is reasonable to assume that these sequences are all independent. Thus, if we consider each linear chromosome to be a block, then we obtain the following bounds when doing adaptive data analysis with in a simple setting:

▶ **Theorem 10.** *Let $\mathcal{M}$ be a query answering mechanism $\mathcal{M}$, such that when given $(X_1, X_2, \dots X_n) \sim D^n$ for a population distribution D such that the attributes are divided into fully independent blocks $B_1, B_2, \dots B_m$, given a data analyst A, $\mathcal{M}$ proceeds as follows:*

- *$\mathcal{M}$ refuses to answer queries that involve attributes in different blocks.*
- *$\mathcal{M}$ ensures that, for each block B_i, the interaction restricted to queries on the block B_i is (ϵ, δ)-DP and (α, β) sample accurate.*

Then, for every $c, d > 0$, $\mathcal{M}$ is (α', β') distributionally accurate where $\alpha' = \alpha + e^{\epsilon} - 1 + c + 2d$ and $\beta' = m\left(\frac{\beta}{c} + \frac{\delta}{d}\right)$.

Proof. Let $X = (X_1, X_2, \dots, X_n)$ denote the sample that $\mathcal{M}$ takes as input. For each i, we conduct a thought experiment to define a query answering mechanism $\mathcal{M}_i'$ as follows:

$\mathcal{M}_i'$ takes as data the i^{th} block of X (which we denote $X^{(i)}$). Then, $\mathcal{M}_i'$ samples new values for blocks $B_1, B_2, \dots, B_{i-1}, B_{i+1}, \dots, B_m$ from D[3]. Let X' denote this new sample. $\mathcal{M}_i'$ then interacts with an analyst A by running $\mathcal{M}$ with the new sample X'. The queries on any block other than B_i update the states of A and $\mathcal{M}_i'$, but are not considered to be queries and answers of the interaction between A and $\mathcal{M}_i'$.

Now, by definition, when A interacts with $\mathcal{M}_i'$, only queries on the i^{th} block interact with the data in any way, which means this interaction is (ϵ, δ)-DP. Furthermore, it is (α, β)-sample accurate from the assumption that $\mathcal{M}$ was (α, β)-sample accurate for the queries on block B_i. Thus, by theorem 3.5 from [10], $\mathcal{M}_i'$ is $(\alpha', \beta'/m)$ distributionally accurate.

Now, since B_i is independent from all other blocks, $X' \sim D^n$. Thus, all $\mathcal{M}_i'$ does is interact with A as if it were $\mathcal{M}$ on sample X', except it only writes queries on block B_i on the transcript. When we consider the distribution with randomness over the choice

[3] The reason why this is just a thought experiment is that in reality the mechanism will not know the distribution D. This is why we carry out data analysis in the first place.

of sample, the mechanism, and the adversary, the distribution of transcripts produced by Interact$(\mathcal{M}'_i, A, X^{(i)})$ is therefore exactly the same as the distribution of transcripts produced by Interact$(\mathcal{M}, A, X)$, with the added postprocessing step of throwing away every query and answer asked about some block other than B_i.

Thus, the distribution of transcripts produced by Interact$(\mathcal{M}, A, X)$ is identical to the distribution of the concatenation of the transcripts of Interact$(\mathcal{M}'_i, A_i, X^{(i)})$ for every i where all of the A_i are copies of A. Taking a union bound over the accuracy guarantees for the latter, we get that $\mathcal{M}$ is (α', β') accurate. ◀

4 Partial Independence

This model is a generalization of the previous model, as the intuition that attributes which are close to one another can be related produces data which do not satisfy the assumptions necessary for the full independence model (consider items that are close, but on different sides of a block boundary). We therefore generalize our result to the case where adjacent blocks are allowed to be related. Additionally, we restrict access to the data to a streaming model. This allows us to achieve stronger accuracy guarantees; specifically, we obtain a bound with twice the privacy loss of full independence; without the streaming restriction it would be thrice the privacy loss.

To do this, we first introduce the following lemma that we will use in the proof. Intuitively, the lemma states that a transformation of individuals preserves privacy.

▶ **Lemma 11.** *Let $\mathcal{M}^{\mathcal{Y}}$ be an (ε, δ)-differentially private mechanism with data domain $\mathcal{Y}$. Then the mechanism $\mathcal{M}^{\mathcal{X}}$, defined next and having data domain $\mathcal{X}$, is also (ε, δ)-differentially private.*

$\mathcal{M}^{\mathcal{X}}$ takes as input a database $X \in \mathcal{X}^n$ and constructs $Y = f(X) \in \mathcal{Y}^n$, where f is a randomized mapping $f : \mathcal{X} \to \mathcal{Y}$. The randomness is chosen independently every time f is called, and we define $Y = f(X) = \{f(x) \mid x \in X\}$. Then, $\mathcal{M}^{\mathcal{X}}$ runs $\mathcal{M}^{\mathcal{Y}}$ on Y: given (oracle) access to any adversary A, $\mathcal{M}^{\mathcal{X}}$ simply acts as a channel, conveying queries from A to $\mathcal{M}^{\mathcal{Y}}$ and responses from $\mathcal{M}^{\mathcal{Y}}$ to A.

Proof. Fix an adversary A, and let Π be the random variable denoting the transcript of the interaction between A and $\mathcal{M}^{\mathcal{Y}}$; that is, $\Pi \in \text{Interact}(\mathcal{M}^{\mathcal{Y}}, A, *)$, the set of all transcripts that can be produced by these two parties.

Let Q be a random variable that represents the value of the database given to $\mathcal{M}^{\mathcal{Y}}$ by $\mathcal{M}^{\mathcal{X}}$, with randomness over X and f. Since $\mathcal{M}^{\mathcal{Y}}$ is (ε, δ)-differentially private we have that, for any event $E \in \text{Interact}(\mathcal{M}^{\mathcal{Y}}, A, *)$ and any Y' adjacent to Y,

$$\Pr[\Pi \in E \mid Q = Y] \leq e^{\epsilon} \Pr[\Pi \in E \mid Q = Y'] + \delta,$$

where the probabilities are over the randomness of $\mathcal{M}^{\mathcal{Y}}$ and A.

Fix an adjacent pair X and X' in $\mathcal{X}^n$ and let i be the index in which they differ. For $R \in \{X, X'\}$ we have:

$$\Pr[\Pi \in E \mid R_i = X_i] = \sum_{y \in \mathcal{Y}} \Pr[\Pi \in E \mid Q_i = y] \cdot \Pr[Q_i = y \mid R_i = X_i]$$

since the event $\Pi \in E$ is independent of the original database R conditional on the transformed database Y. Here the probabilities are over the randomness in the mapping f and the randomness in the $[\mathcal{M}^{\mathcal{Y}}, A]$ interaction, i.e., the coin flips of $\mathcal{M}^{\mathcal{Y}}$ and A.

Let y^* denote the outcome which minimizes $\Pr[\Pi \in E \mid Q_i = y^*]$. Additionally, recall that we defined Y as the input to $\mathcal{M}^{\mathcal{Y}}$, so if we fix $Y_i = y$, then $Y = (f(X_{-i}), y)$.

$$\Pr_{\text{Interact}(\mathcal{M}^{\mathcal{X}}, A, X)}[\Pi \in E \mid R = X] \tag{1}$$

$$= \sum_{y \in \mathcal{Y}} \Pr_{f(X_{-i}), \text{Interact}(\mathcal{M}^{\mathcal{Y}}, A, Y)}[\Pi \in E \mid Q_i = y] \cdot \Pr_{f(X_i)}[Q_i = y \mid R_i = X_i] \tag{2}$$

$$\leq \sum_{y} (e^{\epsilon} \Pr[\Pi \in E \mid Q_i = y^*] + \delta) \cdot \Pr[Q_i = y \mid R_i = X_i] \tag{3}$$

$$= (e^{\epsilon} \Pr[\Pi \in E \mid Q_i = y^*] + \delta) \sum_{y} \Pr[Q_i = y \mid R_i = X_i] \tag{4}$$

$$= (e^{\epsilon} \Pr[\Pi \in E \mid Q_i = y^*] + \delta) \sum_{y} \Pr[Q_i = y \mid R_i = X_i'] \tag{5}$$

$$\leq \sum_{y} (e^{\epsilon} \Pr[\Pi \in E \mid Q_i = y] + \delta) \Pr[Q_i = y \mid R_i = X_i'] \tag{6}$$

$$= \delta + e^{\epsilon} \sum_{y} \Pr[\Pi \in E \mid Q_i = y] \Pr[Q_i = y \mid R_i = X_i'] \tag{7}$$

$$= e^{\epsilon} \Pr[\Pi \in E \mid R = X'] + \delta \tag{8}$$

Since Y_{-i} is sampled independently from Y_i and X_i, the inequality in line (3) holds when we condition on any value of Y_{-i} by definition of (ϵ, δ)-DP, so it must also hold when we take the probability over Y_{-i} as well. The equality in line (5) follows by the law of total probability. ◀

▶ **Theorem 12.** *Suppose we have a query answering mechanism $\mathcal{M}$, such that when given $(X_1, X_2, \ldots X_n) \sim D^n$ for a population distribution D where the attributes are grouped into 1-dependent blocks $\{B_1, B_2, \ldots B_m\}$(sequences of consecutive attributes), and a stateful data analyst A, $\mathcal{M}$ procedes as follows:*

At each time step $t \in [m]$, $\mathcal{M}$ has an arbitrary (ϵ, δ)-DP interaction with A in which A asks linear queries about block B_t and $\mathcal{M}$ answers the queries in such a way that the interaction is (α, β) sample accurate. The transcript is denoted by S_t.

Then, for every $c, d > 0$, $\mathcal{M}$ is (α', β') accurate where $\alpha' = \alpha + e^{2\epsilon} - 1 + c + 2d$ and $\beta' = m\left(\frac{\beta}{c} + \frac{2\delta}{d}\right)$.

Proof. First, for each $i \in [m]$, we define a query answering mechanism $\mathcal{M}_i'$ and adversary A_i' as follows:

$\mathcal{M}_i'$ takes as input the i^{th} block of our original sample of n individuals $(X_1, X_2, \ldots X_n) \sim D^n$, which we will denote $X^{(i)}$. It then resamples the first $i-1$ blocks from D^n conditional on $X^{(i)}$. We will refer to this database of the $i-1$ resampled blocks and the i^{th} block as Y. Then, A_i' and $\mathcal{M}_i'$ run Interact$(\mathcal{M}, A, Y)$ for t from 1 to i, and we denote the transcript generated at time t by this interaction as S_i'. While both parties may keep track of $S_1', S_2', \ldots S_{i-1}'$, only $S_i = S_i'$ is considered to be the transcript of this interaction.

Now, we note that the distribution of transcripts $S_1, S_2, \ldots, S_i$ produced by $\mathcal{M}_i'$ and $\mathcal{M}$ are identical. This is because, analogously to the proof of theorem 10, first sampling a block and then sampling the rest of the data conditional on that block produces the same distribution as sampling all of the data at once.

Now, we shall analyze the accuracy of $\mathcal{M}_i'$. By definition, the first $i-2$ blocks are independent of $X^{(i)}$, so the part of Interact$(\mathcal{M}_i', A_i', X^{(i)})$ that generates $S_1', S_2', \ldots S_{i-2}'$ is independent of $X^{(i)}$ and thus does not incur any privacy loss with respect to $X^{(i)}$.

For S_{i-1}, recall that $(X_1, X_2, \ldots, X_n)$ are drawn from the distribution iid. Thus, when we fix $X^{(i)}$ and resample the $i-1^{st}$ block conditional on $X^{(i)}$, the value of the $i-1^{st}$ block of each individual X_j is a randomized mapping of the i^{th} block the same individual X_j, independent of every other individual $X_{j'}$. Then, the interaction between $\mathcal{M}'_i$ and A'_i on block B_{i-1} is (ϵ, δ)-DP with respect to the resample $i-1^{st}$ block. Thus, by Lemma 11, the part of Interact$(\mathcal{M}'_i, A'_i, X^{(i)})$ that generates S_{i-1} is (ϵ, δ)-DP.

Finally, because $\mathcal{M}$ is (ϵ, δ)-DP on the interaction in each block, the part of Interact$(\mathcal{M}'_i, A'_i, X^{(i)})$ that generates S_i is (ϵ, δ)-DP. As such, $\mathcal{M}'_i$ is $(2\epsilon, 2\delta)$-DP and (α, β)-sample accurate. By theorem 3.5 from [10], $\mathcal{M}'_i$ is $(\alpha', \beta'/m)$ distributionally accurate.

This tells us that $\mathcal{M}'_i$ is $(\alpha', \beta'/m)$ distributionally accurate for each i, and just like in Theorem 10, we can concatenate the transcripts S_i computed from $\mathcal{M}'_i$ for each i to get the transcript $S_1, S_2, \ldots, S_m$ with the same distribution as the interaction between $\mathcal{M}$ and A. taking a union bound over the probabilities of failure over these m mechanisms tells us that $\mathcal{M}$ is (α', β') distributionally accurate. ◀

5 Exponential Decay

Our final model directly captures the idea that the strength of the relationship between two attributes should be decreasing with the distance between them. We model this via following definition:

▶ **Definition 13.** *In the decaying correlation model with parameter p, we are given attributes $B_1, B_2, \ldots B_n$, such that for each i, B_i and B_{i+1} are independent with probability p, and otherwise they are arbitrarily related. The event of B_i and B_{i+1} being related and B_j and B_{j+1} being related are independent for all $i \neq j$, and for any $i < j$, B_i and B_j are related iff $B_{i'-1}$ is related to $B_{i'}$ for every $i < i' \leq j$.*

With this model, there is some dependence between all of the attributes. However, due to the way it is defined, the dependence only exists with small probability over the sample between distant attributes. Thus, we can utilize similar arguments as above, and simply add this small probability to the probability of failure.

▶ **Theorem 14** (General Access)**.** *Given a database X in the decaying correlation model with parameter p and m attributes, a mechanism $\mathcal{M}$ which satisfies the following properties while interacting with an adversary A is (α', β')-distributionally accurate where for all integers $d > 0$:*

$$\alpha' = \alpha + (e^{\epsilon} - 1) + c + 2f, \quad \beta' = m\left(\frac{\beta}{c} + \frac{\delta}{f}\right) + 2n(1-p)^{d+1}.$$

1. *For each i, $\mathcal{M}$ restricted to queries that involve at least one of the attributes $\{B_{i-2d}, B_{i-2d+1}, \ldots B_{i+2d}\}$ is (ϵ, δ)-DP.*
2. *For each i, $\mathcal{M}$ restricted to queries that involve only attributes in the set $\{B_{i-d}, B_{i-d+1}, \ldots B_{i+d}\}$ is (α, β) sample accurate.*
3. *Any query can only involve attributes B_i and B_j if $|i-j| \leq d$.*

Proof. Let D be the population distribution. For each i, we define a query answering mechanism $\mathcal{M}'_i$ as follows:

$\mathcal{M}'_i$ takes as data the attributes $\{B_{i-d}, \ldots, B_{i+d}\}$ of n individuals $(X_1, X_2, \ldots X_n) \sim D^n$, which we shall refer to as $X^{(i)}$. $\mathcal{M}'_i$ then constructs Y by sampling the attributes $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i-d-1}, B_{i+d+1}, \ldots, B_{i+2d}\}$ for n individuals from the population D conditional on agreeing with $X^{(i)}$ on the attributes $\{B_{i-d}, \ldots, B_{i+d}\}$. The rest of the attributes for these n individuals are sampled from D independently from $X^{(i)}$.

Then, $\mathcal{M}'_i$ interacts with an adversary A by simulating $\mathcal{M}$ on the dataset Y. Any query which asks about an attribute outside of the set $\{B_{i-d}, \ldots, B_{i+d}\}$ still takes place in the interaction, but it is not recorded in the transcript.

This construction guarantees that our (α, β)-sample accuracy bound on $\mathcal{M}$ restricted to queries that involve at least one of the attributes $\{B_{i-d}, B_{i-d+1}, \ldots, B_{i+d}\}$ also applies to $\mathcal{M}'_i$, since $\{B_{i-d}, \ldots, B_{i+d}\}$ are exactly the attributes $\mathcal{M}'_i$ takes as data, so sample accuracy is well-defined over these queries.

The privacy loss of $\mathcal{M}'_i$ can be bounded by the privacy loss when we only consider queries that involve at least one of the attributes $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i+2d}\}$ since all of the other attributes are sampled independently from the data. We are given that this is $(\epsilon, \delta) - DP$.

Thus, $\mathcal{M}'_i$ is $(\epsilon, \delta) - DP$ and (α, β)-sample accurate. By the transfer theorem, $\mathcal{M}'_i$ on the set of queries involving attribute B_i is (α', β_2)-distributionally accurate for

$$\alpha' = \alpha + (e^\epsilon - 1) + c + 2f, \quad \beta_2 = \frac{\beta}{c} + \frac{\delta}{f}.$$

Now, by construction, if we condition on $Y \sim X$, we can get the same distribution of transcripts as $\text{Interact}(\mathcal{M}'_i, A, X^{(i)})$ by computing the transcript of $\text{Interact}(\mathcal{M}, A, X)$ restricted to queries that involve only attributes in the set $\{B_{i-d}, B_{i-d+1}, \ldots, B_{i+d}\}$. Additionally, by assumption 2, we know that the guarantee for $\mathcal{M}'_i$ applies to every query that involves attribute B_i. As such, (α', β_2) bounds the distributional accuracy of all queries involving attribute B_i in $\text{Interact}(\mathcal{M}, A, X)$. Thus, we can bound the distributional accuracy of $\mathcal{M}$ by union bounding the probability that the distributional error of any answer in any of $\{\mathcal{M}'_1, \mathcal{M}'_2, \ldots \mathcal{M}'_m\}$ is greater than α', conditional on $Y \sim X$.

We get $Y \sim X$ iff X satisfies the property that all attributes outside of $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i-2d}\}$ are independent from all attributes in the set $\{B_{i-d}, \ldots, B_{i+d}\}$. This happens iff B_{i-2d-1} is independent from B_{i-d} and B_{i+2d+1} is independent from B_{i+d} for every individual in X. This probability is at least $1 - 2n(1-p)^{d+1}$ by taking a union bound over the 2 attributes B_{i-2d-1} and B_{i+2d+1} for each of the n individuals.

As such we can bound the accuracy of the answers $\mathcal{M}$ produces to the queries involving some attribute in the set $\{B_{i-d}, B_{i-d+1}, \ldots B_{i+d}\}$ by simply adding the probability that it does not produce the same distribution of transcripts as $\mathcal{M}'_i$ to the probability of failure, so it is (α', β')-distributionally accurate for

$$\beta' = m\beta_2 + 2n(1-p)^{d+1}$$

or equivalently,

$$\alpha' = \alpha + (e^\epsilon - 1) + c + 2f, \quad \beta' = m\left(\frac{\beta}{c} + \frac{\delta}{f}\right) + 2n(1-p)^{d+1}$$

as desired. ◀

We can improve the parameters by constraining access to the *sliding window* model studied in other contexts (see, for example, the tutorial [9] on sliding window aggregation algorithms, and the references therein). Details may be found in the appendix.

6 Using the Label in the Mechanism

In this Section, we show that, at a small cost in accuracy, we can extend our results to analyses that incorporate the labels. This is a pleasant surprise, as the labels are "morally" exposed to high privacy loss. The key idea to note here is that even though we use the exact marginal

distribution of the label, which cannot be done privately, the query-answering mechanisms that we use as sub-processes take data without the label, for which no information has been revealed to the adversary.

▶ **Theorem 15.** *Suppose the following is true:*

1. *There is a binary attribute y which we refer to as the "label."*
2. *We have a mechanism $\mathcal{M}_0$ which is (α_0, β_0)-distributionally accurate when $y = 0$ for every individual in the distribution.*
3. *We have a mechanism $\mathcal{M}_1$ which is (α_1, β_1)-distributionally accurate when $y = 1$ for every individual in the distribution.*

Now, consider the mechanism $\mathcal{M}$ which on input S, runs as follows:

1. *Partition S into samples $S_0 = \{s \in S \mid s \text{ has } y = 0\}$ and $S_1 = \{s \in S \mid s \text{ has } y = 1\}$*
2. *When $\mathcal{M}$ receives query q from the adversary, it asks q to $\mathcal{M}_0$ on sample S_0 and gets answer a_0. It then asks q to $\mathcal{M}_1$ on sample S_1 and gets answer a_1. $\mathcal{M}$ then returns the answer*

$$a_0 \frac{|S_0|}{|S|} + a_1 \frac{|S_1|}{|S|}.$$

Let D be the population distribution, D_y be the marginal distribution of the label y, and $p = \Pr_{y \sim D_y}[y = 0]$. Then, $\mathcal{M}$ is (α, β)-distributionally accurate for any $\delta > 0$ and

$$\alpha = p\alpha_0 + (1-p)\alpha_1 + \frac{\delta p}{\sqrt{n}}, \quad \beta = \beta_0 + \beta_1 + 2e^{-2\delta^2}.$$

Proof. To approximate the population proportion, we want to take p times the output of $\mathcal{M}_0$ plus $1 - p$ times the output of $\mathcal{M}_1$. To see this, if we let D_0 be the population distribution when we let $y = 0$, and D_1 be the population distribution when we let $y = 1$, then we have for any query q, $pq(D_0) + (1-p)q(D_1) = q(D)$. Thus, for query q_j, if we let a_j be the answer from $\mathcal{M}_0$ and a'_j be the answer from $\mathcal{M}_1$, we have

$$\begin{aligned} |pa_j + (1-p)a'_j - q_j(D)| &= \left|p\left(a_j - q_j(D_0)\right) + (1-p)\left(a'_j - q_j(D_1)\right)\right| \\ &\leq p|a_j - q_j(D_0)| + (1-p)|a'_j - q_j(D_1)|. \end{aligned}$$

Now, if we let $\hat{p} = \frac{|S_0|}{|S|}$, then we have by the triangle inequality

$$\begin{aligned} |\hat{p}a_j + (1-\hat{p})a'_j - q_j(D)| &\leq |\hat{p}a_j + (1-\hat{p})a'_j - pa_j - (1-p)a'_j| + |pa_j + (1-p)a'_j - q_j(D)| \\ &\leq |(\hat{p} - p)(a_j - a'_j)| + p|a_j - q_j(D_0)| + (1-p)|a'_j - q_j(D_1)| \\ &\leq |(\hat{p} - p)| + p|a_j - q_j(D_0)| + (1-p)|a'_j - q_j(D_1)| \end{aligned}$$

where the last inequality comes from the fact that the answers are bounded betweeen $[0, 1]$. Now, $\hat{p} \sim \frac{1}{n}\text{binom}(n, p)$, so we can apply Chernoff to get that for any $\delta > 0$,

$$\Pr\left[|p - \hat{p}| < \frac{\delta p}{\sqrt{n}}\right] < 2e^{-2\delta^2}.$$

Furthermore, by assumption, we know that $|a_j - q_j(D_0)| \leq \alpha_1$ with probability $1 - \beta_1$, and $|a'_j - q_j(D_1)| \leq \alpha_2$ with probability $1 - \beta_2$. Thus, taking a union bound, we get that for any $\delta > 0$, $\mathcal{M}$ is (α, β)-sample accurate for

$$\alpha = p\alpha_0 + (1-p)\alpha_1 + \frac{\delta p}{\sqrt{n}}, \quad \beta = \beta_0 + \beta_1 + 2e^{-2\delta^2}. \qquad \blacktriangleleft$$

7 Discussion

It is common practice in other fields to consider restricted classes of adversaries, where it is often possible to obtain better bounds. For example, while Byzantine Agreement requires $n \geq 3t + 1$ processors if the number of arbitrary failures can be as large as t, it requires only $n \geq t + 1$ processors to handle t fail-stop faults. Similarly, in cryptographic protocols the bounds for *honest-but-curious* adversaries are often better than for the case of processors that diverge arbitrarily from the protocol.

This history, combined with the fact that an algorithm that only protects benign data analysts could still be of use, naturally leads to the question of whether it is possible to get better accuracy/adaptivity tradeoffs for more benign adaptive accuracy adversaries. Efforts to define an appropriate class of benign failure modes were stymied, however, by Freedman's paradox, which states that when we have a dataset of n individuals and n attributes, all of which are independent of a label y, we will find some attribute which is strongly correlated with y with high probability. We feel this gives an example of a very natural error, naïve but not malicious [7].

Our conclusion is that some restriction – e.g., on data models or access models – is therefore required, which led to this work. It would be interesting to find other natural restrictions that lead to improvements comparable to – or better than – those obtained in this work.

References

1 Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. *SIAM Journal on Computing*, 50(3):STOC16–377, 2021.

2 Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013.

3 Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM Journal on Computing*, 47(5):1888–1938, 2018.

4 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *Advances in Neural Information Processing Systems*, 28, 2015.

5 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. *arXiv e-prints*, 2014. `arXiv:1411.2664`.

6 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126, 2015.

7 David A. Freedman. A note on screening regression equations. *The American Statistician*, 37(2):152–155, 1983. URL: `http://www.jstor.org/stable/2685877`.

8 Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70. IEEE, 2010.

9 Martin Hirzel, Scott Schneider, and Kanat Tangwongsan. Sliding-window aggregation algorithms: Tutorial. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, pages 11–14, 2017.

10 Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees, 2019. `arXiv:1909.03577`.

11 Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.
12 Montgomery Slatkin. Linkage disequilibrium – understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9:477–485, 2008.

A Sliding Window Model for Exponential Decay

▶ Remark 16. The form of this bound looks mostly identical to the bound in Theorem 14, with a slightly better probability of failure. However, one must note that the privacy guarantee is now restricted to the set $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i+d}\}$ rather than $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i+2d}\}$ as it was before, so this does in fact give us a multiplicative constant improvement over Theorem 14.

▶ **Theorem 17** (Sliding Window)**.** *Given a database X in the decaying correlation model with parameter p and m attributes, a mechanism $\mathcal{M}$ which satisfies the following properties while interacting with an adversary A is (α', β')-distributionally accurate where*

$$\alpha' = \alpha + (e^{\epsilon} - 1) + c + 2f, \quad \beta' = m\left(\frac{\beta}{c} + \frac{\delta}{f}\right) + n(1-p)^{d+1}.$$

1. *For each i, $\mathcal{M}$ restricted to queries that involve only attributes in the set $\{B_{i-2d}, B_{i-2d+1}, \ldots B_{i+d}\}$ is (ϵ, δ)-DP.*
2. *For each i, $\mathcal{M}$ restricted to queries that involve B_i is (α, β)-sample accurate.*
3. *Any query can only involve attributes B_i and B_j if $|i - j| \leq d$.*
4. *After answering a query involving attribute B_i, the mechanism can no longer answer queries involving attributes $B_1, B_2, \ldots B_{i-d}$.*

Proof. We define $X^{(i)}$ and $\mathcal{M}'_i$ as in theorem 14, except we now stop the interaction immediately after A asks the first query which involves an attribute in the set $\{B_{i+d+1}, \ldots, B_m\}$ and before $\mathcal{M}'_i$ answers.

This interaction still contains every query which involves attribute B_i by assumption 4, and these queries are all well-defined by assumption 3, so analogously to in theorem 14, $\mathcal{M}'_i$ is (α, β)-sample accurate.

This time, the privacy loss of $\mathcal{M}'_i$ can be bounded by the privacy loss when we only consider queries that involve the attributes $\{B_{i-2d}, B_{i-2d+1}, \ldots, B_{i+d}\}$ since there are no queries asked about $\{B_{i+d}, B_{i+d+1}, \ldots, B_{i+2d}\}$. We are given that this is $(\epsilon, \delta) - DP$.

Thus, $\mathcal{M}'_i$ is $(\epsilon, \delta) - DP$ and (α, β)-sample accurate on all the queries in the transcript. Hence, by the transfer theorem, $\mathcal{M}'_i$ on the set of queries involving attribute B_i is (α', β_2)-distributionally accurate for

$$\alpha' = \alpha + (e^{\epsilon} - 1) + c + 2f, \quad \beta_2 = \frac{\beta}{c} + \frac{\delta}{f}.$$

In this setting, we cannot have any query involving $\{B_{i+d+1}, \ldots, B_m\}$ be answered by $\mathcal{M}'_i$ or by $\mathcal{M}$ prior to any query involving B_i. Hence, this time, we note that the probability that some attribute in $\{B_1, B_2, \ldots, B_{i-2d-1}\}$ is related to B_{i-d} is at most $n(1-p)^{d+1}$ by taking a union bound over the n individuals, in which case Interact$(\mathcal{M}'_i, A, X^{(i)})$ restricted to queries that involve attribute B_i produces the same distribution of transcripts as Interact$(\mathcal{M}, A, X)$ restricted to queries that involve attribute B_i.

As such, similarly to in Theorem 14, we can bound the accuracy of the answers in Interact$(\mathcal{M}, A, X)$ by adding the probability that X has some attribute in $\{B_1, B_2, \ldots, B_{i-2d-1}\}$ related to B_{i-d} to the probability that any Interact$(\mathcal{M}'_i, A, i)$ has an answer with error greater than α. Thus, it is (α', β')-distributionally accurate for

$$\alpha' = \alpha + (e^{\epsilon} - 1) + c + 2f, \quad \beta' = m\left(\frac{\beta}{c} + \frac{\delta}{f}\right) + n(1-p)^{d+1}.$$ ◀

Differential Secrecy for Distributed Data and Applications to Robust Differentially Secure Vector Summation

Kunal Talwar
Apple, Cupertino, CA, USA

Abstract

Computing the noisy sum of real-valued vectors is an important primitive in differentially private learning and statistics. In private federated learning applications, these vectors are held by client devices, leading to a distributed summation problem. Standard Secure Multiparty Computation protocols for this problem are susceptible to poisoning attacks, where a client may have a large influence on the sum, without being detected.

In this work, we propose a poisoning-robust private summation protocol in the multiple-server setting, recently studied in PRIO [14]. We present a protocol for vector summation that verifies that the Euclidean norm of each contribution is approximately bounded. We show that by relaxing the security constraint in SMC to a differential privacy like guarantee, one can improve over PRIO in terms of communication requirements as well as the client-side computation. Unlike SMC algorithms that inevitably cast integers to elements of a large finite field, our algorithms work over integers/reals, which may allow for additional efficiencies.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols

Keywords and phrases Zero Knowledge, Secure Summation, Differential Privacy

Digital Object Identifier 10.4230/LIPIcs.FORC.2022.7

Acknowledgements We would like to thank Ulfar Erlingsson for many helpful discussions, and the anonymous referees for their feedback.

1 Introduction

We investigate the problem of distributed private summation of a set of real vectors, each of Euclidean norm at most 1. Each client device holds one of these vectors and the goal is to allow a server to compute the sum of these vectors. Privacy constraints require that an adversary not learn too much about any of these vectors, and this constraint will be expressed as a differential privacy [16] requirement.

This is a common primitive to private federated learning and statistics. In a setting of a trusted server, the clients could send the vectors to the server, which could then output the sum with appropriate noise added to ensure differential privacy. A natural solution then is to use tools from secure multiparty computation to simulate this trusted server. This approach goes back to the early days of differential privacy [15], and has been heavily investigated [11, 8]. Practical protocols applying this approach have to deal with clients dropping out during the protocol, and often scale poorly with the number of clients. The security guarantee of SMC ensures that we learn nothing except the (noisy) sum. However, a malicious client in many of these protocols can contribute a vector with arbitrarily large norm and go completely undetected. Addressing this manipulability would require additional modification to these protocols, making them less feasible.

An elegant way out is possible under slightly stronger trust assumptions. Corrigan-Gibbs and Boneh [14] show that if we have a set of S servers where at least one of them is trusted, we can efficiently get both privacy and integrity[1]. In our application, this framework gives a protocol that validates that each vector has norm at most 1, and computes the sum of vectors. The security guarantee here says that other than the output and the fact that the inputs have norm at most 1, any strict subset of servers learns nothing about the clients' inputs. If the clients add a small amount of noise to their inputs, or more generally, use a local randomizer, the final output can be shown to be differentially private. As long as all the inputs are bounded in norm, the validity predicates are all 1 and hence have no information. The overall security guarantee then says that the view of any strict subset of servers is differentially private with respect to the input vectors.

Note that perfect secrecy here is impossible as the output itself leaks information about the inputs. In the approach described above, *What we compute* does not leak too much about the input since we are computing a differentially private output. *How we compute it*, i.e. the computation protocol itself provides perfect secrecy *subject to* the output.

Our guarantee of interest is the leakage about any input from the process as well as the output, i.e. the sum of the privacy costs from the *what* and the *how*. In this work, we relax the secrecy guarantee of the protocol to a differential secrecy guarantee. We show that this allows for simpler and more efficient algorithms for the robust vector summation problem.

As a warm-up, we first show a natural variant of secret sharing that satisfies differential secrecy. We next show that one can privately verify that theof a secret-shared vector is bounded, if one allows some slack. We present a simple protocol based on random projections. Our protocol accepts all vectors of norm at most 1 with high probability. Additionally, a vector with too large a norm (polylogarithmic in the parameters) will be rejected with high probability. Thus we have some robustness: a malicious client can affect the sum by more than norm 1, but not arbitrarily more. Our privacy proof here relies on a new result on the privacy bounds for noisy random projections. Unusually for a differential privacy result, here we exploit the randomness of the "query". Compared to PRIO, our verification algorithm requires no additional work from clients, and requires less communication between servers.

With secret-sharing and norm-verification over secret shares in place, our algorithm for summation is simple. The clients secret-share their vectors, and the servers run the norm-verification protocol on all the clients. For the clients that pass the norm verification, each server adds up their secret shares. The servers now hold additive secret shares of the summation, which can be communicated between servers to derive the vector summation.

This then eliminates the need for the client to perform any additional computation ($\Theta(d)$ in PRIO) or communication ($\Theta(\sqrt{d})$ [9] in PRIO). The validity check comes at zero cost to the client. This comes at a small increase in the inter-server communication from 3 field elements to a logarithmic number of real numbers.

Our algorithms can work over real numbers or integers, instead of finite fields. Compressing these to reduce communication, for example by truncating or rounding does not affect the privacy guarantee, allowing one to find a representation that provides an acceptable tradeoff between accuracy and communication cost.

In practice, as we discuss in Section 7, this can be a significant saving, especially in settings such as federated learning where the vectors being aggregated are high-dimensional gradients and the client to server communication is often the bottleneck. For typical parameters, where PRIO would need a large finite field needing 128 bits per coordinate (or at the very least 32 bits per coordinate), using real numbers can bring us down to 8 or 16 bits per coordinate.

[1] We defer the precise definitions to Section 3

Several natural questions remain. Our norm verification, and hence our robustness guarantee for summation, is approximate. We reject vectors with large enough norm. It would be interesting to reduce, or even eliminate this approximation, while maintaining the efficiency advantages of our protocol. Given the practical relevance of robust summation, it would also be compelling to improve distributed proofs of norm bound in the standard PRIO setting.

Finally, relaxing perfect secrecy in secure multiparty computation, or more broadly in cryptography to differential secrecy may allow for more efficient protocols in other settings.

2 Related Work

The question of simultaneously studying the differentially private function (the *What*) and the cryptographic protocol for computing it (the *How*) was first studied by Beimel, Nissim and Omri [5]. They showed that in the Secure Function Evaluation (SFE) setting without a trusted server, one can provably gain in efficiency of the protocol for summing 0-1 values. This differential privacy-based definition of security was subsequently used by Backes et al. [2], who show that this relaxation allows one to use imperfect randomness in certain cryptographic protocols.

Private anonymous summation protocols using mutliple servers go back to at least the split-and-mix protocol of Ishai et al. [23]. In the context of differential privacy, these have gained a lot of importance given recent results in the shuffle model of privacy [7, 17, 13, 4]. Recent works [3, 20, 19] have improved the efficiency of these results. These protocols however suffer from the manipulability issue: it is easy for one malicious client to significantly poison the sum without getting detected.

Another line of work [8] proposes practical secure summation protocol under different trust assumptions. These protocols also suffer from the manipulability problem. Recent works such as [29, 6] address the scaling challenges in that work.

The two-party version of some of these questions have been studied by [27, 22]. Kairouz, Oh and Vishwanathan [25, 24] study private secure multiparty computation under a local differential privacy constraint. In a different vein, Cheu, Smith and Ullman [12] show that locally differentially private algorithms are fairly manipulable by small subsets of users, and quantify their manipulability.

3 Definitions

We would like the protocol to satisfy several properties. We define appropriate notions of these first.

▶ **Definition 1** (Completeness). *A protocol* Π *is* $(1-\beta)$*-complete w.r.t.* $\mathcal{L}$ *if for all* $x \in \mathcal{L}$*, the protocol accepts* x *with probability at least* $(1-\beta)$*.*

▶ **Definition 2** (Soundness). *A protocol* Π *is* β*-sound w.r.t.* $\mathcal{L}$ *if for* $x \notin \mathcal{L}$*, the protocol accepts with probability at most* β*.*

Let $\mathcal{L}_r$ denote the set of vectors with $\|\cdot\|_2$ norm at most r. We will show completeness w.r.t. $\mathcal{L}_1$ and soundness w.r.t. $\mathcal{L}_\rho$. for a parameter $\rho > 1$.

Additionally, we would like a mild relaxation of Zero Knowledge, inspired and motivated by the notion of Differential Privacy. We first recall a notion of near-indistinguishability used in Differential Privacy:

▶ **Definition 3.** *Two random variables P and Q are said to be (ε, δ)-close, denoted by $P \approx_{(\varepsilon,\delta)} Q$ if for all events S, $Pr[P \in S] \leq \exp(\varepsilon) \cdot \Pr[Q \in S] + \delta$, and similarly, $Pr[Q \in S] \leq \exp(\varepsilon) \cdot \Pr[P \in S] + \delta$*

One can relax the secrecy requirements in cryptography to differential secrecy. Here we define this notion for Zero Knowledge[2].

▶ **Definition 4.** *We say a protocol Π is (ε, δ)-Differentially Zero Knowledge w.r.t. $\mathcal{L}$ if there is a distribution Q such that for all $x \in \mathcal{L}$, the distribution $\Pi(x)$ of the protocol's transcript on input x satisfies $\Pi(x) \approx_{(\varepsilon,\delta)} Q$.*

Note that here we require privacy, or differential zero knowledge for $x \in \mathcal{L}$. While one can naturally define a computational version of this definition, along the lines of computational differential privacy definitions [28], we restrict ourselves to the information-theoretic version in this work.

In this work, we will be using multi-verifier protocols. Here the notion of near Zero Knowledge is with respect to a strict subset of verifiers. The definition here is the simplest one where the prover starts with an input x and verifiers start with no input, as this will suffice for our purpose. It can naturally be extended to other setups, for example when the verifiers already hold some shares of the input and a witness.

▶ **Definition 5.** *A single-prover, multiple-verifier protocol Π is (ε, δ)-Differentially Zero Knowledge w.r.t to a subset T of parties if there is a distribution Q dependent only on inputs of T and the output of the protocol, such that for any input $x \in \mathcal{L}$, the distribution of messages from T^c to T is (ε, δ)-close to Q.*

Attack Models. In our work, the client will play the role of the prover, and the servers will play the role of the verifiers. We interchangeably use client/server and prover/verifier terminology as appropriate. We will prove completeness and privacy for honest-but-curious prover. We will establish soundness against an arbitrary malicious provers. This implies that a client that is behaving according to the protocol will get a strong privacy guarantee, and will be accepted with high probability. A malicious client will still likely be caught, and may not get a privacy assurance. Our protocols will have privacy against an a strict subset of servers being malicious, as long as at least one of the servers is honest. The soundness and completeness results will assume that all servers are honest. Thus some subsets of servers behaving maliciously can hurt the utility of the protocol, but not the privacy.

For the robust aggregation problem, we argue the privacy of the process given the final sum, and leave to a different argument the question of the privacy of the sum itself. The protocol can easily be modified by adding noise to the sum to ensure that the sum itself is private; this can be done by having each server add sufficient noise, or by implementing a distributed noise-addition protocol. Our modular approach allows us to separately analyze the privacy cost of the output of the protocol. In particular, we may apply different analyses depending on whether we consider distributed noise addition, or apply local randomizers and rely on privacy amplification by shuffling. We defer additional discussion to Section 7.

[2] This is the *local DP* version of ZK which is appropriate in this setting. One can similarly define a central DP version, where the simulator has access to all but one client's input

Secure Summation

The secure summation problem is defined as follows. There is a set of N clients with client i holding a vector $\mathbf{x}_i \in \mathbb{R}^d$ with $\|\mathbf{x}_i\| \leq 1$. Our goal is to design a protocol with S servers such that for suitable parameters $\varepsilon, \delta, \rho, \beta$, the following properties hold:

Correctness: When all parties are honest, the protocol allows a designated server to compute a vector $\mathbf{y} \in \mathbb{R}^d$ such that $\mathbf{y} = \sum_i \mathbf{x}_i$ with probability at least $(1-\beta)$.

Privacy: For any honest client i, the protocol is (ε, δ)-Differentially Zero Knowledge w.r.t. any subset of parties that excludes at least one server.

Robustness: For any possibly malicious client i, the computed summation $\mathbf{y}$ differs from the output $\mathbf{y}_{-i}$ without client i in norm by at most ρ, i.e. $\|\mathbf{y} - \mathbf{y}_{-i}\|_2 \leq \rho$, except with probability at most β.

In words, we would like a protocol that is private w.r.t. to any honest client as long as at least one of the S servers is honest. Thus an honest client that trusts at least one of the servers to be honest is assured of a differential privacy guarantee. The robustness property gives an integrity guarantee if all servers are honest. The parameter $\rho \geq 1$ controls how much any client can impact the output of the protocol. Note that a malicious client can always behave as if their input was $\mathbf{x}'_i$ for any arbitrary vector of norm 1. The robustness requirement here puts an upper bound on how much a malicious client can distort the summation The correctness and robustness properties will allow failure with probability β. Depending on the application, a small constant β may be acceptable.

4 Preliminaries

We state two important properties of the differential privacy notion of closeness.

▶ **Proposition 6.** *Suppose that* $P \approx_{(\varepsilon,\delta)} Q$ *and* $P' \approx_{(\varepsilon',\delta')} Q'$. *Then*

Post Processing: *For any function* f, $f(P) \approx_{(\varepsilon,\delta)} f(Q)$.

Simple Composition: $(P, P') \approx_{(\varepsilon+\varepsilon',\delta+\delta')} (Q, Q')$.

The following is a restatement of the privacy of the Gaussian mechanism [16, Thm A.1].

▶ **Lemma 7.** *Let* $\varepsilon, \delta > 0$ *and let* $\mathbf{x} \in \mathbb{R}^d$ *satisfy* $\|\mathbf{x}\|_2 \leq 1$. *Let* $P \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_d)$ *and let* $Q \sim \mathbf{x} + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_d)$. *Then* $P \approx_{(\varepsilon,\delta)} Q$ *if* $\sigma \geq 2\sqrt{\ln \frac{2}{\delta}}/\varepsilon$.

We next prove the following simple result on the privacy properties of noisy random projections.

▶ **Lemma 8.** *Let* G *be a random matrix in* $\mathbb{R}^{k \times d}$ *such that for a constant* c_δ, *every* $\mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\| \leq 1$ *satisfies*

$$\Pr[\|G\mathbf{x}\| \geq c_\delta] \leq \delta, \tag{1}$$

where the probability is taken over the distribution of G. *Let* $\sigma = 2c_\delta\sqrt{\ln \frac{2}{\delta}}/\varepsilon$. *Then for any* $\mathbf{x} \in \mathbb{R}^d$ *with* $\mathbf{x} \leq 1$,

$$(G, \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_k)) \approx_{(\varepsilon,2\delta)} (G, G\mathbf{x} + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_k)).$$

Proof. Fix $\mathbf{x}$ and let $\mathcal{E}$ be the event that $\|G\mathbf{x}\| \geq c_\delta$. By Lemma 7, we have that conditioned on the event $\mathcal{E}$,

$$(G, \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_k)) \approx_{(\varepsilon,\delta)} (G, G\mathbf{x} + \mathcal{N}(\mathbf{0}, \sigma^2 \mathbb{I}_k)).$$

By Equation (1), $\Pr[\mathcal{E}] \geq 1-\delta$. The claim now follow from the definition of (ε, δ)-closeness. ◀

We next recall a version of the Johnson-Lindenstrauss lemma on the length of random projections.

▶ **Lemma 9** (Gaussian Ensemble JL). *Let $G \in \mathbb{R}^{k\times d}$ be a random matrix where each $G_{ij} \sim \mathcal{N}(0, \frac{1}{k})$. Then for any $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\| \le 1$,*

$$\Pr[\|G\mathbf{x}\| \notin (1 \pm O(\sqrt{(\ln \tfrac{1}{\delta})/k}))\|\mathbf{x}\|] \le \delta$$

To get more precise estimates, we recall that the sum of squares of k $\mathcal{N}(0, \frac{1}{k})$ random variables is distributed as a (scaled version of a) chi-square distribution χ^2_k. We will use the following tail bounds for χ^2_k random variables from Laurent and Massart [26, Lemma 1 rephrased]:

▶ **Theorem 10.** *Let Q be a χ^2_k random variable. Then for any $\beta > 0$,*

$$\Pr[\frac{1}{k}Q \le 1 - 2\sqrt{x/k}] \le \exp(-x),$$
$$\Pr[\frac{1}{k}Q \ge 1 + 2\sqrt{x/k} + 2x/k] \le \exp(-x).$$

Combining Theorem 10 with Lemma 8, we get the following useful corollary.

▶ **Corollary 11.** *Let $G \in \mathbb{R}^{k\times d}$ be a random matrix where each $G_{ij} \sim \mathcal{N}(0, \frac{1}{k})$ and let $c_\delta = \sqrt{1 + 2\sqrt{(\ln \frac{1}{\delta})/k} + 2(\ln \frac{1}{\delta})/k}$. Let $\sigma = 2c_\delta\sqrt{\ln \frac{2}{\delta}}/\varepsilon$. Then for any $\mathbf{x} \in \mathbb{R}^d$ with $\mathbf{x} \le 1$,*

$$(G, \mathcal{N}(\mathbf{0}, \sigma^2\mathbb{I}_k)) \approx_{(\varepsilon,2\delta)} (G, G\mathbf{x} + \mathcal{N}(\mathbf{0}, \sigma^2\mathbb{I}_k)).$$

5 Warm-up: Secret Sharing Real-valued Vectors

As a prelude to our result on norm verification, we first show how the standard secret sharing protocol extends to real-valued vectors, when allowing for Differential secrecy. Consider the protocol for secret-sharing a real-valued vector of norm at most 1 between S servers shown in Algorithm 1.

Algorithm 1 Secret Sharing a real vector.

Prover($\mathbf{x}$):
  Input: Vector $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\| \le 1$.
  Parameters: $\sigma_{SS} \in \mathbb{R}$.
  Generate $\mathbf{g}_1, \ldots, \mathbf{g}_{S-1} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{SS}\mathbb{I}_d)$ using private randomness.
  Send $\mathbf{x} - \sum_{i=1}^{S-1} \mathbf{g}_i$ to Verifier 0.
  for $i = 1 \ldots S-1$ **do**
    Send $\mathbf{g}_i$ to Verifier i.

To prove the differential secrecy for this protocol, we show a simulator for any subset of verifiers in Algorithm 2.

We next argue that this secret sharing scheme is differentially secure.

▶ **Theorem 12.** *Fix any $T \subsetneq [S]$. Then Prover($\mathbf{x}$)$|_T \approx_{(\varepsilon,\delta)}$ Simulator(T) for $(S - |T|)\sigma^2_{SS} \ge 4\ln \frac{2}{\delta}/\varepsilon^2$.*

Algorithm 2 Simulator for Algorithm 1.

Simulator($T \subsetneq [S]$)**:**
  Input: T proper subset of S
  Parameters: $\sigma_{SS} \in \mathbb{R}$.
  for $i \in T$ **do**
    if $i \neq 0$ **then**
      Generate $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \sigma_{SS}^2 \mathbb{I}_d)$.
      Send $\mathbf{g}_i$ to Verifier i.
  if $0 \in T$ **then**
    Generate $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, (S - |T|)\sigma_{SS}^2 \mathbb{I}_d)$.
    Send $\mathbf{g} - \sum_{i \in T; i \neq 1} \mathbf{g}_i$ to Verifier 0.

Proof. If $0 \notin T$, the simulation is perfect: indeed each verifier in T receives an independent Gaussian vector with variance $\sigma_{SS}^2 \mathbb{I}_d$ in both distributions. When $0 \in T$, consider the distribution of the message to Verifier 0 conditioned on $T \setminus [0]$.

The simulator output to Verifier 0 is distributed as $\mathcal{N}(-\sum_{i \in T; i \neq -0} \mathbf{g}_i, (S - |T|)\sigma_{SS}^2 \mathbb{I}_d)$. The message to verifier 0 from the prover, conditioned on $\{\mathbf{g}_i\}_{i \in T: i \neq 0}$ is distributed as $\mathcal{N}(\mathbf{x} - \sum_{i \in T; i \neq 0} \mathbf{g}_i, (S - |T|)\sigma_{SS}^2 \mathbb{I}_d)$. The claim now follows from the privacy of the Gaussian mechanism (Lemma 7). ◀

The differential secrecy implies that an honest prover's privacy is protected against an arbitrary collusion of verifiers short of all of them. Note also that by making σ_{SS} larger, we can improve the privacy cost. A larger σ_{SS} only costs us in terms of the precision to which these messages should be communicated to ensure that the sum of secret shares is close to $\mathbf{x}$. Note that we can post-process these vectors (both in the algorithm and its simulation), e.g. by rounding or truncation. By the post-processing property of differential privacy, the differential secrecy is maintained.

6 Differential Zero Knowledge Proofs of bounded norm

We next describe our DZK protocol to verify a Euclidean norm bound. The first step is to secret-share the vector between the two verifiers as in the previous section. The rest of the protocol only involves the verifiers; the prover code therefore is identical to secret-sharing.

The second step is norm estimation and happens amongst the verifiers. As a first cut, suppose that the servers aggregate their shares, while adding noise to each share to preserve privacy. This would require adding d-dimensional gaussian noise to each share. This noise being fresh and independent will contribute to the norm of the computed sum, which will now be about $\sqrt{d}$, and will have variance growing polynomially with d. This will make it impossible to estimate the norm better than some polynomial in d, and thus our gap ρ will grow polynomially with the dimension.

To improve on this, we will use random projection into a k-dimensional space for a parameter k independent of the dimension. Being a lower-dimensional object, a projection can be privately estimated much more accurately. The choice of the projection dimension k will give us a trade-off between the privacy parameters and the gap assumption. Intuitively, we rely on the Johnson-Lindenstrauss lemma, which says that the Euclidean norm of a vector is approximately preserved under random projections. Since projection is a linear operator, computing the projection of a secret-shared vector is straight-forward. Verifier 0 here takes the special role of collecting an estimate of a random projection of $\mathbf{x}$, computing its norm and sharing the Accept/Reject bit.

Algorithm 3 Protocol for Norm Verification.

Input: Prover has a vector $\mathbf{x} \in \mathbb{R}^d$
Output: Verifiers must agree on Accept.

Prover($\mathbf{x}$*)*:
- **Input:** Vector $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\| \leq 1$.
- **Parameters:** $\sigma_{SS} \in \mathbb{R}$.
- Generate $\mathbf{g}_1, \ldots, \mathbf{g}_{S-1} \sim \mathcal{N}(\mathbf{0}, \sigma_{SS}^2 \mathbb{I}_d)$ using private randomness.
- Send $\mathbf{x} - \sum_{i=1}^{S-1} \mathbf{g}_i$ to Verifier 0.
- **for** $i = 1 \ldots S-1$ **do**
  - Send $\mathbf{g}_i$ to Verifier i.

Verifier-0:
- **Parameters:** Integer k. Threshold $\tau \in \mathbb{R}$.
- Receive $\mathbf{z}_0$ from Prover. `// Expected to be` $\mathbf{x} - \sum_{i=1}^{S-1} \mathbf{g}_i$
- Generate $\mathbf{W} \in \mathbb{R}^{k \times d}$ with each $W_{ij} \sim \mathcal{N}(0, \frac{1}{k})$ using private randomness.
  `// This version assumes honest Verifier 0. To allow malicious Verifier 0,` W `is generated using randomness shared amongst verifiers.`
- Send $\mathbf{W}$ to Verifiers $1, \ldots, S-1$.
- **for** $i = 1 \ldots S-1$ **do**
  - Receive $\mathbf{y}_i$ from Verifier i. `// Expect` $\mathbf{y}_i = \mathbf{W}\mathbf{g}_i + \mathit{Noise}$.
- Compute $\mathbf{v} = \mathbf{W}\mathbf{z}_0 + \sum_{i=1}^{S-1} \mathbf{y}_i + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)$. `// Expect` $\mathbf{v} = \mathbf{W}\mathbf{x} + \mathit{Noise}$.
- **if** $|\mathbf{v}| \geq \tau$ **then**
  - Accept $= 0$
- **else**
  - Accept $= 1$
- Send Accept to Verifiers $1, \ldots, S-1$.

Verifier-i $(i \geq 1)$:
- **Parameters:** Integer k. Noise scale $\sigma_v \in \mathbb{R}$.
- Receive $\mathbf{z}_i$ from Prover. `// Expected to be` $\mathbf{g}_i$
- Receive $\mathbf{W} \in \mathbb{R}^{k \times d}$ from Verifier-0.
- Compute $\mathbf{y}_i = \mathbf{W}\mathbf{z}_i + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)$.
- Send $\mathbf{y}_i$ to Verifier-0.
- Receive Accept from Verifier-0.

We start with establishing compeleteness, which will determine the acceptance threshold τ. We will then show soundness for an appropriate ρ.

▶ **Theorem 13** (Completeness). *Suppose that the prover and the verifiers are honest and the* $\|\mathbf{x}\| \leq 1$. *Then for* $\tau \geq \sqrt{(\frac{1}{k} + |S|\sigma_v^2)(k + 2\ln\frac{1}{\beta} + 2\sqrt{k \ln \frac{1}{\beta}})}$,

$$\Pr[\text{Accept} = 1] \geq 1 - \beta.$$

Proof. Under the assumptions, $W\mathbf{x}$ is distributed as $\mathcal{N}(0, \frac{\|\mathbf{x}\|_2^2}{k}\mathbb{I})$. The noise added by each server is distributed as $\mathcal{N}(0, \sigma_v^2 \mathbb{I})$, and all of these Gaussian random variables are independent. Thus $\mathbf{v}$ computed by Verifier 0 is distributed as $\mathcal{N}(0, (\frac{\|\mathbf{x}\|_2^2}{k} + |S|\sigma_v^2)\mathbb{I})$, and its squared norm is distributed as $(\frac{\|\mathbf{x}\|_2^2}{k} + |S|\sigma_v^2)Q$, where Q is a χ_k^2 random variable. Thus

$$\Pr[\|\mathbf{v}\|_2^2 \geq \tau^2] = \Pr[Q \geq (\frac{1}{k} + |S|\sigma_v^2)^{-1}\tau^2].$$

Plugging the upper tail bounds from Theorem 10, the result follows. ◀

▶ **Theorem 14** (Soundness). *Suppose that the verifiers are honest and suppose that* $\|\sum_{i=0}^{S-1} z_i\| \geq \rho$, *where* z_i *is the message to verifier* i. *Then for* $\rho^2 \geq \frac{k\tau^2}{k-2\sqrt{k\ln\frac{1}{\beta}}} - k|S|\sigma_v^2$,

$$\Pr[\mathsf{Accept} = 1] \leq \beta.$$

Proof. As in the proof of Theorem 13, now $\|\mathbf{v}\|_2^2$ is distributed as $(\frac{\rho^2}{k} + |S|\sigma_v^2)Q$ for a χ_k^2 random variable Q. Using the lower tail bounds from Theorem 10, it suffices to ensure

$$(\frac{\rho^2}{k} + |S|\sigma_v^2)(k - 2\sqrt{k\ln\frac{1}{\beta}}) \geq \tau^2.$$

Rearranging, the claim follows. ◀

Some discussion on k is in order. A small k ensures that we need to add less noise and thus get better estimates. At the same time, larger k ensures stronger concentration of the χ_k^2 random variable. For intuition, we next estimate the bound on ρ^2 from Theorem 14, plugging in τ from Theorem 13. Setting $\lambda = \frac{\sqrt{\ln\frac{1}{\beta}}}{k}$ and assuming λ is small enough, we can write

$$\begin{aligned}
\rho^2 &= \frac{k\tau^2}{k - 2\sqrt{k\ln\frac{1}{\beta}}} - k|S|\sigma_v^2 \\
&= (1 + k|S|\sigma_v^2)\frac{k + 2\ln\frac{1}{\beta} + 2\sqrt{k\ln\frac{1}{\beta}}}{k - 2\sqrt{k\ln\frac{1}{\beta}}} - k|S|\sigma_v^2 \\
&= (1 + k|S|\sigma_v^2)\frac{1 + 2\lambda + 2\sqrt{\lambda}}{1 - 2\sqrt{\lambda}} - k|S|\sigma_v^2 \\
&\approx (1 + k|S|\sigma_v^2)(1 + O(\sqrt{\lambda})) - k|S|\sigma_v^2 \\
&= 1 + O(k|S|\sigma_v^2\sqrt{\lambda}) \\
&\approx 1 + O(|S|\sigma_v^2 k^{\frac{1}{2}}(\ln\frac{1}{\beta})^{\frac{1}{4}}).
\end{aligned}$$

Taking $k = \Theta(\sqrt{\ln\frac{1}{\beta}})$ suffices to ensure λ is small enough for the approximations above to be valid. This leads to $\rho^2 = \Theta(|S|\sigma_v^2\sqrt{\ln\frac{1}{\beta}})$. In practice, one may want to use the exact cdf for the χ_k^2 distribution instead of the tail bounds used in the theorems.

We now prove the differential zero knowledge property of the algorithm. We assume that verifier 0 is honest. We will then relax this assumption using shared randomness.

▶ **Theorem 15** (DZK assuming honest Verifier 0). *Suppose that* $\|\mathbf{x}\|_2 \leq 1$. *If the prover and Verifier-0 are honest, then for any* $T \subset [S] \setminus \{0\}$, T*'s view is* (ε, δ)*-DZK as long as* $\sigma_v \geq 2c_\delta\sqrt{\ln\frac{4}{\delta}}/\varepsilon$.

Proof. The simulator is defined in Algorithm 4. The simulator sends messages to verifiers in T in steps 4, 6, and 13. The messages in steps 4 and 6 follows exactly the same distribution as that in the mechanism, with all $\mathbf{g}_i$'s and the matrix $\mathbf{W}$ being independent normal. The message in step 13 is the Accept bit, which is computed as a post-processing of the vector

Algorithm 4 Simulator for Algorithm 3.

1 *Simulator*($T \subsetneq [S]; 0 \notin T$)**:**
  Input: T proper subset of S
  Parameters: $\sigma_{SS}, \sigma_v, \tau \in \mathbb{R}$, integer k.
2 **for** $i \in T$ **do**
3 Generate $\mathbf{g}_i \sim \mathcal{N}(\mathbf{0}, \sigma_{SS}^2 \mathbb{I}_d)$.
4 Send $\mathbf{g}_i$ to Verifier i.
5 Generate $\mathbf{W} \in \mathbb{R}^{k \times d}$ with each $W_{ij} \sim \mathcal{N}(0, \frac{1}{k})$.
6 Send $\mathbf{W}$ to each Verifier in T.
7 Receive $\{\mathbf{y}_i\}_{i \in T}$.
8 Compute $\mathbf{v}_{Sim} = \sum_{i \in T}(\mathbf{y}_i - \mathbf{W}\mathbf{g}_i) + \mathcal{N}(\mathbf{0}, (S - |T|)\sigma_v^2 \mathbb{I}_k)$.
9 **if** $|\mathbf{v}_{Sim}| \geq \tau$ **then**
10 Accept $= 0$
11 **else**
12 Accept $= 1$
13 Send Accept to all verifiers.

$\mathbf{v}_{Sim}$ computed in step 8. The corresponding Accept bit in the protocol is obtained by the same post-processing of $\mathbf{v}$ computed by Verifier 0 in step 6. Since the prover is honest, we can write:

$$\begin{aligned}
(\mathbf{W}, \mathbf{v}) &= (\mathbf{W}, \mathbf{W}\mathbf{z}_0 + \sum_{i=1}^{S-1} \mathbf{y}_i + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)) \\
&= (\mathbf{W}, \mathbf{W}(\mathbf{x} - \sum_{i=1}^{S-1} \mathbf{g}_i) + \sum_{i=1}^{S-1} \mathbf{y}_i + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)) \\
&= (\mathbf{W}, \mathbf{W}(\mathbf{x} - \sum_{i=1}^{S-1} \mathbf{g}_i) + \sum_{i \in T; i \neq 0} \mathbf{y}_i + \sum_{i \notin T; i \neq 0} \mathbf{y}_i + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)) \\
&= (\mathbf{W}, \mathbf{W}\mathbf{x} + \sum_{i \in T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i) + \sum_{i \notin T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i) + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)).
\end{aligned}$$

Since provers outside of T follow the protocol,

$$\begin{aligned}
(\mathbf{W}, \mathbf{v}) &= (\mathbf{W}, \mathbf{W}\mathbf{x} + \sum_{i \in T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i) + \sum_{i \notin T; i \neq 0} \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k) + \mathcal{N}(0, \sigma_v^2 \mathbb{I}_k)) \\
&= (\mathbf{W}, \mathbf{W}\mathbf{x} + \sum_{i \in T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i) + \mathcal{N}(0, (S - |T|)\sigma_v^2 \mathbb{I}_k)) \\
&= (\mathbf{W}, (\mathbf{W}\mathbf{x} + \mathcal{N}(0, (S - |T|)\sigma_v^2 \mathbb{I}_k)) + \sum_{i \in T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i)) \\
&\approx_{(\varepsilon, \delta)} (\mathbf{W}, \mathcal{N}(0, (S - |T|)\sigma_v^2 \mathbb{I}_k) + \sum_{i \in T; i \neq 0} (\mathbf{y}_i - \mathbf{W}\mathbf{g}_i)) \\
&= (\mathbf{W}, \mathbf{v}_{Sim}).
\end{aligned}$$

Here we have used Corollary 11 in the second to last step. ◀

The honest prover assumption is necessary to give privacy to the prover. The assumption on Verifier 0 being honest is necessary as well in the protocol as stated: a malicious Verifier 0 that can choose an adversarial $\mathbf{W}$ can violate the privacy constraint. For example, a verifier

that knows that the true $\mathbf{x}$ lies in a certain k-dimensional subspace can choose the projection matrix $\mathbf{W}$ to project to that subspace. This will make the projected vector to have length much larger than 1, and invalidate the assumptions in Lemma 8. We next show that this is the only place where we need Verifier 0 to be honest. Thus given a distributed oracle for randomly selecting $\mathbf{W}$, e.g. using shared randomness, we have privacy as long as one of the Verifiers is honest.

▶ **Theorem 16** (DZK assuming randomly chosen $\mathbf{W}$). *Suppose that* $\|\mathbf{x}\|_2 \leq 1$. *Further suppose that the prover is honest and the matrix* $\mathbf{W}$ *shared in Step 4 by Verifier* 0 *is uniformly random. Then for any* $T \subsetneq [S]$, *T's view is* $(\varepsilon + \varepsilon', \delta + \delta')$*-DZK as long as* $\sigma_v \geq 2c_\delta\sqrt{\ln \frac{4}{\delta}}/\varepsilon$ *and* $\sigma_{SS} \geq 2\sqrt{\ln \frac{2}{\delta'}}/\varepsilon'$.

Proof. The proof is nearly identical to the previous proof. When $0 \notin T$, the theorem follows from Theorem 15. When Verifier 0 is in the set, the secret sharing itself is (ε', δ')-DZK, by Theorem 12. The rest of the protocol is (ε, δ)-DZK by repeating the proof of Theorem 15. The result follows. ◀

7 Application to Robust Secure Aggregation

Our protocol for robust secure aggregation (Algorithm 5) builds on the additive secret shares with norm bound verification. The prover part of the protocol is nearly identical to secret sharing, with the only change being that the client sends its identifier j with all the shares. We assume that each client has a unique identifier, though this assumption can be easily relaxed by having the client send a random nonce instead of its identifier.

The verifiers execute the norm verification protocol for each client. Verifier 0 constructs the set of indices J^* that pass the norm verification and shares it with all the verifiers. The verifiers optionally check that J^* is large enough; this part is not needed for our summation protocol, but can be useful to ensuring that the sum itself is differentially private. The verifiers now add up the secret shares for the provers in J^* and share the sum with Verifier 0, that adds up the sums of secret shares to derive the sum.

Algorithm 5 Client Protocol for Robust Secure Aggregation.

Input: Prover j has a vector $\mathbf{x}_j \in \mathbb{R}^d$
Output: Verifiers compute $\sum_j \mathbf{x}_j$

Prover$_j(\mathbf{x}_j)$**:**
  Input: Vector $\mathbf{x}_j \in \mathbb{R}^d$ with $\|\mathbf{x}_j\| \leq 1$.
  Parameters: $\sigma_{SS} \in \mathbb{R}$.
  Generate $\mathbf{g}_1, \ldots, \mathbf{g}_{S-1} \sim \mathcal{N}(\mathbf{0}, \sigma_{SS}^2 \mathbb{I}_d)$ using private randomness.
  Send $\mathbf{x}_j - \sum_{i=1}^{S-1} \mathbf{g}_i$ to Verifier 0.
  for $i = 1 \ldots S-1$ **do**
    Send $(j, \mathbf{g}_i)$ to Verifier i.

The privacy proof is nearly identical to the last section. Indeed up to the computation of J^*, the protocol is exactly equivalent to the norm verification protocol. Verifiers other than verifier 0 do not receive any additional message after J^*, so that a simulator for a subset of verifiers excluding verifier 0 is essentially identical to that in the previous section. Verifier 0 receives a set of vectors $\{s_i\}$. For $i \neq T$, the simulator simulates $s_i \sim \mathcal{N}(\mathbf{0}, |J^*|\sigma_{SS}^2 \mathbb{I}_d)$ subject to the sum of all s_i's being equal to the output. It can be easily verified that this part of the simulation is exact. Privacy follows.

Algorithm 6 Server Protocol for Robust Secure Aggregation.

Input: Prover j has a vector $\mathbf{x}_j \in \mathbb{R}^d$
Output: Verifiers compute $\sum_j \mathbf{x}_j$

Verifier-0:
- **Parameters:** Integer k. Threshold $\tau \in \mathbb{R}$.
- Receive $V_0 = \{(j, \mathbf{z}_0^j)\}$ from Provers. Let $J_0 = \{j : (j, \mathbf{z}_0^j) \in V_0\}$.
- Generate $\mathbf{W} \in \mathbb{R}^{k \times d}$ with each $W_{ij} \sim \mathcal{N}(0, \frac{1}{k})$ using private randomness.
- Send $\mathbf{W}$ to Verifiers $1, \ldots, S-1$.
- **for** $i = 1, \ldots, S-1$ **do**
  - Receive $V_i = \{(j, \mathbf{y}_i^j)\}$ from Verifier i. Let $J_i = \{j : (j, \mathbf{y}_i^j) \in V_i\}$.
- Let $J = \cap_i J_i$.
- **for** $j \in J$ **do**
  - Compute $\mathbf{v}^j = \mathbf{W}\mathbf{z}_0^j + \sum_{i=1}^{S-1} \mathbf{y}_i^j + \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbb{I}_k)$.
  - **if** $|\mathbf{v}^j| \le \tau$ **then**
    - add j to J^*; `// J* collects j that pass the norm verification.`
- Send J^* to Verifiers $1, \ldots, S-1$.
- *Optional:* **if** *not* *Valid*(J^*) **then**
  - Abort; `// Ensure J* is large enough.`
- $\mathbf{s}_0 = \mathbf{0}$.
- **for** $j \in J^*$ **do**
  - $\mathbf{s}_0 = \mathbf{s}_0 + \mathbf{z}_0^j$.
- **for** $i = 1, \ldots, S-1$ **do**
  - Receive $\mathbf{s}_i$ from Verifier i.
- Return $\sum_{i=0}^{S-1} \mathbf{s}_i$.

Verifier-i ($i \ge 1$):
- **Parameters:** Integer k. Noise scale $\sigma_v \in \mathbb{R}$.
- Receive $V_i = \{j, \mathbf{z}_i^j)\}$ from Provers. Let $J_i = \{j : (j, \mathbf{z}_i^j) \in V_i\}$.
- Receive $\mathbf{W} \in \mathbb{R}^{k \times d}$ from Verifier-0.
- **for** $j \in J_i$ **do**
  - Compute $\mathbf{y}_i^j = \mathbf{W}\mathbf{z}_i^j + \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbb{I}_k)$.
- Send $\{(j, \mathbf{y}_i^j)\}$ to Verifier-0.
- Receive J^* from Verifier-0.
- *Optional:* **if** *not* *Valid*(J^*) **then**
  - Abort; `// Ensure J* is large enough.`
- $\mathbf{s}_i = \mathbf{0}$.
- **for** $j \in J^*$ **do**
  - $\mathbf{s}_i = \mathbf{s}_i + \mathbf{z}_i^j$.
- Send $\mathbf{s}_i$ to Verifier-0.

We next prove the correctness. We wish to prove that when all the parties are honest, then the sum is correctly computed except with a small failure probability. With probability $1 - n\beta$, each of the n norm verification steps succeed, so that J^* is the set of all clients. Conditioned on this, the correctness of the secret sharing and the commutativity of addition immediately imply that the sum computed by Verifier 0 is the desired sum of all vectors.

We note that for many applications such as gradient accumulation, a weaker correctness notion may suffice. If J^* is a random subset of $[n]$ with each j landing in J^* with probability $(1 - \beta)$, we get an unbiased estimate of the sum. For this weaker definition of correctness, the failure probability does not need to be scaled by a multiplicative factor of n which translates to a smaller threshold τ, and thus better robustness.

Finally we argue robustness. Consider a client j. If the client secret-shares a vector with norm at most ρ, then their affect on the computed sum is clearly at most ρ. On the other hand, if client j's shares add up to a vector of norm larger than ρ, it will be rejected by the norm verification step except with probability β. This means that $j \notin J^*$ and j's secret shares do not contribute at all to the compute sum. Additionally, if j does not send messages to all the verifiers, their input gets rejected as well.

When the validity check on J^* is added, the robustness claim is weaker. Indeed suppose that the validity check compares $|J^*|$ to a threshold, say $\frac{n}{2}$. Then the $(\frac{n}{2}+1)$th malicious client can cause the computation to abort. The robustness guarantee now says that if the computation succeeds, then the effect of any potentially malicious client is bounded. Further, we can argue that a small number of malicious clients cannot cause the computation to abort, except with small probability.

We have thus established correctness, robustness and privacy of our protocol. For n clients sending vectors in $\mathbb{R}^d$, the communication cost for each client is $O(d|S|)$. The communication cost between servers is $O(dk + nk + d|S|)$. Recall that a $k = O(\sqrt{\ln n})$ suffices to get polynomially small completeness and soundness.

On the Privacy of the Sum

We established the privacy of the protocol, conditioned on the sum. How do we ensure the privacy of the sum itself? One option is to add differential privacy noise to the sum itself to ensure privacy. If each verifier adds noise to s_i, we get a differential privacy guarantee against any strict subset of the verifiers. The eventual noise variance for the sum then scales with the number of servers.

An appealing alternative is to distribute the noise generation itself. This approach goes back to Dwork et al. [15]. The question of generating noise on different clients such that the sum has a certain distribution has been studied for this reason. While Gaussian noise has the nice property that sum of gaussians is a gaussian, Laplace noise is also "divisible" [21, 3]. These arguments however require that the summation be done over real numbers. In particular, this means that for privacy to hold, the constituents of the sum may need to be communicated to sufficiently high precision even if the original vectors are $\{0, 1\}$. Works such as [1] address this question of preserving privacy while reducing the communication.

Recent results on privacy amplification by shuffling offer an elegant way out of this cononudrum. The general results in this direction [7, 13, 17, 4, 18] say that local randomizers, when shuffled give strong central differential privacy guarantees. In particular, since summation is a post-processing of shuffling, these results apply to the sum. The privacy-accuracy trade-offs of the shuffle model are very competitive with the central model for many settings [30, 18]. Moreover, in deployments where local randomizers are used for other reasons, this approach avoids adding additional noise.

This ability to post-process without hurting privacy offers additional benefits. The secret-shares themselves can be rounded, truncated, or compressed without hurting privacy. For example, when the input vectors are $\{0, 1\}$, the secret sharing algorithm can use discrete gaussian noise [10], and truncate all secret shares to $[-B, B]$ for a suitable constant B. This does not affect the privacy claim, and the truncation operator is the identity except with a small probability depending on B. The small loss in accuracy due to rare truncation can be analytically or empirically traded-off against the communication cost. As an example $B = 127$ would suffice for encoding each bit as 8 bits, and would ensure that the likelihood of any single bit being distorted, say for $\sigma_{SS} = 20$ is at most 10^{-8}. This may be an acceptable

error rate in applications where randomized response is used to generate the bit vectors. In comparison the field size in PRIO must grow with the number of clients and for typical values, one would use at least 32 bits.

References

1 Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7564–7575. Curran Associates, Inc., 2018. URL: `http://papers.nips.cc/paper/7984-cpsgd-communication-efficient-and-differentially-private-distributed-sgd.pdf`.

2 Michael Backes, Aniket Kate, Sebastian Meiser, and Tim Ruffing. Secrecy without perfect randomness: Cryptography with (bounded) weak sources. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *Applied Cryptography and Network Security*, pages 675–695, Cham, 2015. Springer International Publishing.

3 B. Balle, J. Bell, A. Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. *ArXiv*, abs/2002.00817, 2020.

4 Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 638–667, Cham, 2019. Springer International Publishing.

5 Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 451–468, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

6 James Bell, K. A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. Cryptology ePrint Archive, Report 2020/704, 2020. URL: `https://eprint.iacr.org/2020/704`.

7 Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 441–459, New York, NY, USA, 2017. Association for Computing Machinery. `doi:10.1145/3132747.3132769`.

8 Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery. `doi:10.1145/3133956.3133982`.

9 Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Zero-knowledge proofs on secret-shared data via fully linear pcps. Cryptology ePrint Archive, Report 2019/188, 2019. URL: `https://ia.cr/2019/188`.

10 Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy, 2020. `arXiv:2004.00010`.

11 T. H. Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 200–214, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

12 Albert Cheu, Adam Smith, and Jonathan Ullman. Manipulation attacks in local differential privacy, 2019. `arXiv:1909.09630`.

13 Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 375–403, Cham, 2019. Springer International Publishing.

14 Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 259–282, Boston, MA, 2017. USENIX Association. URL: `https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs`.

15 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer Verlag, May 2006. URL: `https://www.microsoft.com/en-us/research/publication/our-data-ourselves-privacy-via-distributed-noise-generation/`.

16 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

17 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 2468–2479, USA, 2019. Society for Industrial and Applied Mathematics.

18 Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2021. arXiv:2012.12803 [cs.LG].

19 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3692–3701. PMLR, 18–24 July 2021. URL: `https://proceedings.mlr.press/v139/ghazi21a.html`.

20 Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 798–827, Cham, 2020. Springer International Publishing.

21 Slawomir Goryczka and Li Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 14:463–477, 2017.

22 Vipul Goyal, Ilya Mironov, Omkant Pandey, and Amit Sahai. Accuracy-privacy tradeoffs for two-party differentially private protocols. In *CRYPTO*, pages 298–315. Springer, 2013. `doi:10.1007/978-3-642-40041-4_17`.

23 Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography from anonymity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 239–248, 2006.

24 P. Kairouz, S. Oh, and P. Viswanath. Differentially private multi-party computation. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 128–132, March 2016. `doi:10.1109/CISS.2016.7460489`.

25 Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Secure multi-party differential privacy. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2008–2016. Curran Associates, Inc., 2015. URL: `http://papers.nips.cc/paper/6004-secure-multi-party-differential-privacy.pdf`.

26 B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, October 2000. `doi:10.1214/aos/1015957395`.

27 Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *51st Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE, 2010.

28 Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 126–142, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

29 Jinhyun So, Basak Guler, and A. Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning, 2020. arXiv:2002.04156.

30 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation, 2020. arXiv:2001.03618.