2nd Symposium on Foundations of Responsible Computing

FORC 2021, June 9–11, 2021, Virtual Conference

Edited by

Katrina Ligett
Swati Gupta

LIPICS

Editors

Katrina Ligett
Hebrew University of Jerusalem, Israel
katrina@cs.huji.ac.il

Swati Gupta
Georgia Institute of Technology, Atlanta, Georgia, United States
swatig@gatech.edu

ACM Classification 2012
Security and privacy → Privacy-preserving protocols; Theory of computation → Machine learning theory; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Probability and statistics; Social and professional topics → Computing / technology policy; Computing methodologies → Supervised learning; Security and privacy; Applied computing → Law; Applied computing → Voting / election technologies; Computing methodologies → Machine learning; Computing methodologies → Ranking

ISBN 978-3-95977-187-0

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/978-3-95977-187-0.

Publication date
June, 2021

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.2021.0

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

Regular Papers

Preface

The Symposium on Foundations of Responsible Computing (FORC), now in its second 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.

Twenty-four papers were selected to appear at FORC 2021, held virtually due to the COVID-19 pandemic, on June 9–11, 2021. The 24 papers were selected by the program committee, with the help of additional expert reviewers, out of 52 submissions. FORC 2021 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 17 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 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: the Simons Collaboration on the Theory of Algorithmic Fairness and the Harvard Center of Mathematical Sciences and Applications (CSMA) for partial conference support.

Katrina Ligett, Jerusalem
Swati Gupta, Atlanta, Georgia
April 19, 2021

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Privately Answering Counting Queries with Generalized Gaussian Mechanisms

Arun Ganesh
Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, CA, USA

Jiazheng Zhao
Computer Science Department, Stanford University, CA, USA

Abstract

We give the first closed-form privacy guarantees for the Generalized Gaussian mechanism (the mechanism that adds noise x to a vector with probability proportional to $\exp(-(||x||_p/\sigma)^p)$ for some σ, p), in the setting of answering k counting (i.e. sensitivity-1) queries about a database with (ϵ, δ)-differential privacy (in particular, with low ℓ_∞-error). Just using Generalized Gaussian noise, we obtain a mechanism such that if the true answers to the queries are the vector d, the mechanism outputs answers $\tilde{d}$ with the ℓ_∞-error guarantee:

$$\mathbb{E}\left[||\tilde{d} - d||_\infty\right] = O\left(\frac{\sqrt{k \log\log k \log(1/\delta)}}{\epsilon}\right).$$

This matches the error bound of [18], but using a much simpler mechanism. By composing this mechanism with the sparse vector mechanism (generalizing a technique of [18]), we obtain a mechanism improving the $\sqrt{k \log\log k}$ dependence on k to $\sqrt{k \log\log\log k}$, Our main technical contribution is showing that certain powers of Generalized Gaussians, which follow a Generalized Gamma distribution, are sub-gamma.

In subsequent work, the optimal ℓ_∞-error bound of $O(\sqrt{k \log(1/\delta)}/\epsilon)$ has been achieved by [4] and [9] independently. However, the Generalized Gaussian mechanism has some qualitative advantages over the mechanisms used in these papers which may make it of interest to both practitioners and theoreticians, both in the setting of answering counting queries and more generally.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols

Keywords and phrases Differential privacy, counting queries, Generalized Gaussians

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.1

Related Version *Previous Version*: `https://arxiv.org/abs/2010.01457`

Funding *Arun Ganesh*: Supported in part by NSF Award CCF-1535989.

Acknowledgements The inspiration for this project was a suggestion by Kunal Talwar that Generalized Gaussians could achieve the same asymptotic worst-case errors for query response as the mechanism of Steinke and Ullman. In particular, he suggested a proof sketch of a statement similar to Lemma 14 which was the basis for our proof that lemma. We are also thankful to the anonymous reviewers for multiple helpful suggestions on improving the paper.

1 Introduction

A fundamental question in data analysis is to, given a database, release answers to k numerical queries about a database d, balancing the goals of preserving the *privacy* of the individuals whose data comprises the database and preserving the *utility* of the answers to the queries. A standard formal guarantee for privacy is (ϵ, δ)-differential privacy [6, 5]. A mechanism $\mathcal{M}$ that takes database d as input and outputs (a distribution over) answers $\tilde{d}$ to the queries is (ϵ, δ)-differentially private if for any two databases d, d' which differ by only one individual and for any set of outcomes S, we have:

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 1; pp. 1:1–1:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\Pr_{\tilde{d}\sim\mathcal{M}(d)}\left[\tilde{d}\in S\right] \leq e^{\epsilon}\Pr_{\tilde{d}\sim\mathcal{M}(d')}\left[\tilde{d}\in S\right]+\delta. \tag{1}$$

When $\delta = 0$, this property is referred to ϵ-differential privacy. Without loss of generality, we will treat d (resp. $\tilde{d}$) as a k-dimensional vector corresponding to the answers to the queries (resp. the answers outputted by the mechanism). In this paper, we focus on the setting of *counting queries*, i.e. queries for which the presence of each individual in the database affects the answers by at most 1. In turn, throughout the paper we say a mechanism taking vectors in $\mathbb{R}^k$ as input and outputting distributions over $\mathbb{R}^k$ is (ϵ, δ)-differentially private if (1) holds for any two k-dimensional vectors d, d' such that $||d - d'||_\infty \leq 1$ and any subset S of $\mathbb{R}^k$.

To balance the goals of privacy and utility, we seek a mechanism $\mathcal{M}$ that minimizes some objective function of the (distribution of) additive errors $\tilde{d} - d$, while satisfying (1). One natural and well-understood objective function is the ℓ_1-error $||\tilde{d} - d||_1/k$, which gives the average absolute error of the answers to the queries. The well-known and simple *Laplace mechanism* [6], which outputs $\tilde{d} = d + x$ with probability proportional to $\exp(-||x||_1/\sigma)$ for an appropriate value of σ, achieves expected ℓ_1-error of $O(\min\{\sqrt{k\log(1/\delta)}, k\}/\epsilon)$. A line of works on lower bounds [11, 3] culminated in a result of [18] showing this is optimal up to constants.

A less well-understood objective function is the ℓ_∞-error $||\tilde{d} - d||_\infty$, which gives the maximum absolute error of the answers to the queries. The maximum absolute error is of course a more strict objective function than the average absolute error; indeed, the Laplace mechanism only achieves error $O(k \log k/\epsilon)$ and the Gaussian mechanism (which outputs $\tilde{d} = d + x$ with probability proportional to $\exp(-||x||_2^2/\sigma^2)$ for an appropriate value of σ) achieves error $O(\sqrt{k \log k \log(1/\delta)}/\epsilon)$. The first improvements on ℓ_∞-error over the Laplace and Gaussian mechanisms were given by [18][1]. To summarize, the results of that paper (which prior to this paper were all the best known results) are:

- An ϵ-differentially private mechanism satisfying:

$$\Pr_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty \geq O\left(\frac{k}{\epsilon}\right)\right] \leq e^{-\Omega(k)}, \tag{2}$$

  (this matches a lower bound of [10] up to constants).
- An (ϵ, δ)-differentially private mechanism satisfying:

$$\Pr_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty \geq O\left(\frac{\sqrt{k\log\log k\log(1/\delta)}}{\epsilon}\right)\right] \leq e^{-\log^{\Omega(1)} k}. \tag{3}$$

  The mechanism achieving (3) starts by taking the Gaussian mechanism, and then uses the sparse vector mechanism to correct the entries of x with large error in a private manner.
- A lower bound showing any (ϵ, δ)-differentially private mechanism must satisfy:

$$\mathbb{E}_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty\right] \geq \Omega\left(\frac{\sqrt{k\log(1/\delta)}}{\epsilon}\right). \tag{4}$$

[1] Their paper considers the problem setting where queries ask what fraction of n individuals satisfy some property, i.e. queries have sensitivity $1/n$ instead of 1, and the goal is to find the minimum n needed to achieve error at most α. Achieving error Δ with probability $1-\rho$ in our setting is equivalent to needing $n \geq \Delta/\alpha$ to achieve error α with probability $1-\rho$ in their setting.

The additional $\sqrt{\log k}$ term in the Gaussian mechanism's error bound comes from the fact that Gaussians' largest entries are roughly $\sqrt{\log k}$ times larger than their average entries. More generally, if we consider sampling x with probability proportional to $\exp(-(||x||_p/\sigma)^p)$ for some σ, p, the largest entry will be roughly $\log^{1/p} k$ times larger than the average entry. We refer to this distribution as the *Generalized Gaussian with shape p and scale σ*, as is it referred to in e.g. [17]. This leads to a natural question answered in this paper: What error bounds can we get by instead using *Generalized Gaussian mechanisms*?

1.1 Our Results and Techniques

Our first result is as follows:

▶ **Theorem 1.** *For all $1 \leq p \leq \log k$, $\epsilon \leq O(1)$, $\delta \in [2^{-O(k/p)}, 1/k]$, there exists a (ϵ, δ)-differentially private mechanism $\mathcal{M}$ that takes in a vector $d \in \mathbb{R}^k$ and outputs a random $\tilde{d} \in \mathbb{R}^k$ such that for some sufficiently large constant c, and all $t \geq 0$:*

$$\Pr_{\tilde{d} \sim \mathcal{M}_\sigma^p(d)} \left[||\tilde{d} - d||_\infty \geq \frac{ct\sqrt{kp} \log^{1/p} k \sqrt{\log(1/\delta)}}{\epsilon} \right] \leq e^{-t^p \log k}$$

In particular, this implies:

$$\mathbb{E}_{\tilde{d} \sim \mathcal{M}(d)}[||\tilde{d} - d||_\infty] = O\left(\frac{\sqrt{kp} \log^{1/p} k \sqrt{\log(1/\delta)}}{\epsilon} \right).$$

We also have for all $1 \leq q \leq p$:

$$\mathbb{E}_{\tilde{d} \sim \mathcal{M}(d)} \left[\frac{||\tilde{d} - d||_q}{k^{1/q}} \right] = O\left(\frac{\sqrt{kp \log(1/\delta)}}{\epsilon} \right).$$

We note that the lower bound on δ in Theorem 1 can easily be removed: if δ is smaller than $2^{-O(k/p)}$, we can instead use the mechanism achieving (2), which matches the error guarantees of Theorem 1 in this range of δ. The mechanism is simply to add noise from a Generalized Gaussian with shape p and an appropriate scale parameter σ. In our analysis, we arrive at the bounds $c \leq 2094$ and $\sigma \leq 262 \cdot \frac{\sqrt{kp \log(1/\delta)}}{\epsilon}$, although we did not attempt to optimize the constants in favor of a simpler analysis and presentation. We believe the multiplicative constants in both bounds can be substantially improved with a more careful analysis.

Setting $p = \Theta(\log \log k)$, this result matches the asymptotic error bound of (3). However, this result improves on (3) qualitatively. Although the mechanism achieving (3) is already not too complex, the Generalized Gaussian mechanism we use is even simpler, just adding noise from a well-known distribution. Notably, Generalized Gaussian mechanisms retain the property of the Gaussian mechanism that the noise added to each entry of d is independent (unlike the mechanism giving (3), which uses dependent noise), and that the noise has a known closed-form distribution that is easy to sample from[2]. To the best of our knowledge, this is the first analysis giving privacy guarantees for Generalized Gaussian mechanisms besides that in [14]. Even then, [14] does not give any closed-form bounds on the value of σ needed for privacy. This analysis may be of independent interest for other applications where one would normally use the Gaussian mechanism, but may want to use a Generalized Gaussian mechanism with $p > 2$ to trade average-case error guarantees for better worst-case error guarantees.

[2] see e.g. `https://sccn.ucsd.edu/wiki/Generalized_Gaussian_Probability_Density_Function`.

We give a summary of our analysis here; the full analysis is given in Section 2. We first need to determine what value of σ causes the Generalized Gaussian mechanism to be private. Viewing the Generalized Gaussian mechanism as an instance of the exponential mechanism of [15], this reduces to deriving a tail bound on $||x+1||_p^p - ||x||_p^p$ for x sampled from the noise distribution. If p is even this is roughly equal to $p\sum_{j=1}^k x_j^{p-1}$. By a Chernoff bound on the signs of each random variable in the sum, this is roughly tail bounded by the sum of $\sqrt{k\log(1/\delta)}$ of the x_j^{p-1} random variables. These variables are distributed according to a *Generalized Gamma* distribution, which we prove is sub-gamma in Section B. This gives us the desired tail bound, and thus an upper bound on the σ needed to ensure (ϵ,δ)-differential privacy. To prove the error guarantees, we derive tail bounds on the ℓ_p-norm of x sampled from Generalized Gaussian distributions, as well as on the coordinates of points sampled from unit-radius ℓ_p-spheres, the latter of which is done by upper bounding the volume of "sphere caps" of these spheres.

Building on this result, we improve the previous best-known ℓ_∞ error for answering counting queries with (ϵ,δ)-differential privacy:

▶ **Theorem 2.** *For all $\epsilon \leq O(1)$, $\delta \in [2^{-O(k/\log\log\log k)}, 1/k]$, $t \in [0, O(\frac{\log k}{\log\log k})]$, there exists a (ϵ,δ)-differentially private mechanism $\mathcal{M}$ that takes in a vector $d \in \mathbb{R}^k$ and outputs a random $\tilde{d} \in \mathbb{R}^k$ such that for a sufficiently large constant c:*

$$\Pr_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty \geq \frac{ct\sqrt{k\log\log\log k\log(1/\delta)}}{\epsilon}\right] \leq e^{-\log^t k}.$$

In particular, if we choose e.g. $t=2$ we get:

$$\mathbb{E}_{\tilde{d}\sim\mathcal{M}(d)}[||\tilde{d}-d||_\infty] = O\left(\frac{\sqrt{k\log\log\log k\log(1/\delta)}}{\epsilon}\right).$$

Again, the lower bound on δ can easily be removed using the mechanism achieving (2). We arrive at this result by improving upon Generalized Gaussian mechanisms in the same manner [18] improves upon the Gaussian mechanism: After sampling x from a Generalized Gaussian, we apply the sparse vector mechanism to x to get $\tilde{x}$ which satisfies $||x-\tilde{x}||_\infty \ll ||x||_\infty$. We then just output $\tilde{d} = d + x - \tilde{x}$. The full analysis is given in Section 3. Similarly to [18], the major technical component is showing that at least $k/\log^{\Omega(1)} k$ entries of x are small with high probability, which we do using the tail bounds derived in Section 2. This is necessary for the sparse vector mechanism to satisfy that $||x-\tilde{x}||_\infty$ is, roughly speaking, the $(k/\log^{\Omega(1)} k)$-th largest entry of x rather than the largest entry with high probability.

1.2 Subsequent Work and Comparisons

Following our work, [4] and [9] independently gave mechanisms with optimal expected ℓ_∞-error $O(\sqrt{k\log(1/\delta)}/\epsilon)$, quantitatively improving our results. Since in practice $\sqrt{\log\log k}$ is unlikely to be much larger than the constants hidden by the asymptotic notation (e.g., using the natural log, $\sqrt{\log\log k} = 2$ for $k \approx 5\cdot 10^{23}$), the qualitative differences between our results and these two results make our results still of interest to e.g. practitioners. Theorem 1 is our qualitatively more appealing result, and so we highlight the differences with that result in particular. Again, we note that while the explicit constant in Theorem 1 is likely too large to be of practical interest, we believe this constant can be substantially improved with a more refined analysis, hopefully making the mechanism practical.

The result of [4] remarkably uses a bounded noise distribution, and in turn the *maximum* ℓ_∞-error rather than just the average ℓ_∞-error of their mechanism is bounded, in contrast with Generalized Gaussian mechanisms whose maximum ℓ_∞-error is unbounded. However, a bounded noise distribution cannot e.g. satisfy group differential privacy for all group sizes simultaneously, whereas Generalized Gaussian mechanisms can. Also, while both results simply add noise, Generalized Gaussians are more well-studied than the noise distribution of [4] and can be sampled by simplying powering and rescaling samples from Gamma random variables, which should make them easier to implement in practice.

The result of [9] at a high level adds noise and then repeatedly applies the sparse vector mechanism to correct entries with large noise, in contrast to just adding noise. In addition, their result uses arguably even simpler sampling primitives than ours (it only needs to sample Laplace distributions and permutations of lists), but their overall mechanism needs a somewhat more involved iterative approach rather than a one-shot sample. Finally, as presented the resulting noise distribution from their overall mechanism does not have e.g. a closed-form or independent entries which may be desirable.

1.3 Preliminaries

For completeness, we restate the noise distribution of interest here:

▶ **Definition 3.** *The (multivariate)* ***Generalized Gaussian distribution with shape p and scale σ*** *denoted $GGauss(p, \sigma)$, is the distribution over $x \in \mathbb{R}^k$ with probability distribution function (pdf) proportional to $\exp(-(||x||_p/\sigma)^p)$.*

1.3.1 Sub-Gamma Random Variables

The following facts about sub-gamma random variables will be useful in our analysis:

▶ **Definition 4.** *A random variable X is* ***sub-gamma to the right*** *with variance v and scale c if:*

$$\forall \lambda \in (0, 1/c) : \mathbb{E}[\exp(\lambda(X - \mathbb{E}[X]))] \leq \exp\left(\frac{\lambda^2 v}{2(1 - c\lambda)}\right).$$

Here, we use the convention $1/c = \infty$ if $c = 0$. We denote the class of such random variables $\Gamma^+(v, c)$. Similarly, a random variable X is ***sub-gamma to the left*** *with variance v and scale c, if $-X \in \Gamma^+(v, c)$, i.e.:*

$$\forall \lambda \in (0, 1/c) : \mathbb{E}[\exp(\lambda(\mathbb{E}[X] - X))] \leq \exp\left(\frac{\lambda^2 v}{2(1 - c\lambda)}\right).$$

We denote the class of such random variables $\Gamma^-(v, c)$.

We refer the reader to [1] for a textbook reference for this definition and proofs of the following facts.

▶ **Fact 5.** *If for $i \in [n]$ we have a random variable $X_i \in \Gamma^+(v_i, c_i)$, then $X = \sum_{i \in [n]} X_i$ satisfies $X \in \Gamma^+(\sum_{i \in [n]} v_i, \max_{i \in [n]} c_i)$ (and the same relation holds for $\Gamma^-(v, c)$).*

▶ **Lemma 6.** *If $X \in \Gamma^+(v, c)$ then for all $t > 0$:*

$$\Pr[X > \mathbb{E}[X] + \sqrt{2vt} + ct] \leq e^{-t}.$$

Similarly, if $X \in \Gamma^-(v, c)$ then for all $t > 0$:

$$\Pr[X < \mathbb{E}[X] - \sqrt{2vt} - ct] \leq e^{-t}.$$

▶ **Fact 7.** *Let $X \sim Gamma(a)$, i.e. X has pdf satisfying:*

$$p(x) \propto x^{a-1}e^{-x}.$$

Then X satisfies $X \in \Gamma^+(a,1)$ and $X \in \Gamma^-(a,0)$.

1.3.2 Other Probability Facts

We will use the following standard fact to relate distributions of variables to the distributions of their powers:

▶ **Fact 8** (Change of Variables for Powers)**.** *Let X be distributed over $(0,\infty)$ with pdf proportional to $f(x)$. Let Y be the random variable X^c for $c > 0$. Then Y has pdf proportional to $y^{\frac{1}{c}-1}f(y^{\frac{1}{c}})$.*

Finally, we'll use the following standard tail bounds:

▶ **Lemma 9** (Laplace Tail Bound)**.** *Let X be a Laplace random variable with scale b, $Lap(b)$. That is, X has pdf proportional to $\exp(-|x|/b)$. Then we have $\Pr[|x| \geq tb] \leq e^{-t}$.*

▶ **Lemma 10** (Chernoff Bound)**.** *Let $X_1, X_2, \ldots X_k$ be independent Bernoulli random variables. Let $\mu = \mathbb{E}\left[\sum_{i\in[k]} X_i\right]$. Then for $t \in [0,1]$, we have:*

$$\Pr\left[\sum_{i\in[k]} X_i \geq (1+t)\mu\right] \leq \exp\left(-\frac{t^2\mu}{3}\right).$$

2 Generalized Gaussian Mechanisms

In this section, we analyze the Generalized Gaussian mechanism that given database d, samples $x \sim GGauss(p,\sigma)$ and outputs $\tilde{d} = d + x$. We denote this mechanism $\mathcal{M}_\sigma^p$. When $p = 1$ this is the Laplace mechanism, and when $p = 2$ this is the Gaussian mechanism.

2.1 Privacy Guarantees

We first determine what σ is needed to make this mechanism private. We start with the following lemma, which gives a tail bound on the change in the "utility" function $||\tilde{d} - d||_p^p$ if d changes by $\Delta \in [-1,1]^k$:

▶ **Lemma 11.** *Let $x \in \mathbb{R}^k$ be sampled from $GGauss(p,\sigma)$. Then for $4 \leq p \leq \log k$ that is an even integer, $\delta \in [2^{-O(k/p)}, 1/k]$, and any $\Delta \in [-1,1]^k$ we have with probability $1-\delta$:*

$$||x - \Delta||_p^p - ||x||_p^p \leq 32pk^{1/p-1/2}\sqrt{p\log(1/\delta)}||x||_p^{p-1} + 2k^{\frac{p}{2}}p^2.$$

We remark that the requirement that p be an even integer can be dropped by generalizing the proofs in this section appropriately. However, we can reduce proving Theorem 1 for all p to proving it for only even p by rounding p up to the nearest even integer (at the loss of a multiplicative constant of at most $\sqrt{2}$), and only considering even p simplifies the presentation. So, we stick to considering only even p.

Proof. By symmetry of $GGauss(p,\sigma)$ we can assume Δ has all negative entries. Then we have:

$$||x-\Delta||_p^p - ||x||_p^p = \sum_{i=1}^{k}((x_i-\Delta_i)^p - x_i^p)$$

$$= \sum_{i=1}^{k}\int_{x_i}^{x_i-\Delta} py^{p-1}\mathrm{d}y \le \sum_{i=1}^{k}\int_{x_i}^{x_i-\Delta} p(x_i-\Delta_i)^{p-1}\mathrm{d}y \le p\sum_{i=1}^{k}(x_i-\Delta_i)^{p-1} \le p\sum_{i=1}^{k}(x_i+1)^{p-1}.$$

We want to replace the terms $(x_i+1)^{p-1}$ with terms x_i^{p-1} since the latter's distribution is more easily analyzed. To do so, we use the following observation:

▶ **Fact 12.** *If $p \le \sqrt{k}/2$:*

- *If $x_i > \sqrt{k}$, then we have $(x_i+1)^{p-1} \le \left(1+\frac{1}{\sqrt{k}}\right)^{p-1} x_j^{p-1} \le \left(1+\frac{2p}{\sqrt{k}}\right) x_j^{p-1}$.*
- *If $|x_i| \le \sqrt{k}$, then we have $(x_i+1)^{p-1} - x_i^{p-1} \le (\sqrt{k}+1)^{p-1} - \sqrt{k}^{p-1} \le 2k^{\frac{p}{2}-1}p$.*
- *If $x_i < -\sqrt{k}$, then we have $(x_i+1)^{p-1} \le \left(1-\frac{1}{\sqrt{k}}\right)^{p-1} x_j^{p-1} \le \left(1-\frac{2p}{\sqrt{k}}\right) x_j^{p-1}$.*

Fact 12 gives:

$$\sum_{i=1}^{k}(x_i+1)^{p-1} \le \left(1-\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i<0} x_i^{p-1} + \left(1+\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i\ge 0} x_i^{p-1} + 2k^{\frac{p}{2}}p.$$

It now suffices to show that:

$$-\left(1-\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i<0}|x_i|^{p-1} + \left(1+\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i\ge 0}|x_i|^{p-1} \le 32k^{1/p-1/2}\sqrt{p\log(1/\delta)}||x||_p^{p-1}, \quad (5)$$

with probability at least $1-\delta$. Note that each x_i is sampled independently with probability proportional to $\exp(-(|x_i|/\sigma)^p)$. Since multiplying x by a constant rescales both sides of (5) by the same multiplicative factor, it suffices to show (5) when each x_i is independently sampled with probability proportional to $\exp(-|x_i|^p)$, i.e. when $\sigma = 1$. By change of variables, $y_i = |x_i|^{p-1}$ is sampled from the distribution with pdf proportional to $y_i^{\frac{1}{p-1}-1}\exp(-y_i^{\frac{p}{p-1}})$. This is the Generalized Gamma random variable with parameters $(\frac{1}{p-1}, \frac{p}{p-1})$, which we denote $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$. We show the following property of this random variable in Appendix B:

▶ **Lemma 13.** *For any $p \ge 4$, let Y be the random variable $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$, let $\mu = \mathbb{E}[Y]$. Then $\mu \in [1/p, 1.2/p), Y \in \Gamma^+(\mu, 1)$, and $Y \in \Gamma^-(\mu, 3/2)$.*

Let k' be the number of positive coordinates in x. A Chernoff bound gives that $k' \le \frac{k}{2} + 3\sqrt{k\log\frac{1}{\delta}}$ with probability $1-\delta/3$. By Lemma 13 and Fact 5 $\sum_{i:x_i<0}|x_i|^{p-1}$ is in $\Gamma^-((k-k')\mu, 3/2)$ and $\sum_{i:x_i\ge 0}|x_i|^{p-1}$ is in $\Gamma^+(k'\mu, 1)$ for μ as defined in Lemma 13. We now apply Lemma 6 with $t = \log(6/\delta)$ to each sum. Since $\delta \ge 2^{-O(k/\sqrt{p})}$, $\log(6/\delta) = O(\sqrt{k\log(1/\delta)/p})$, i.e. we are still in the range of δ for which the square-root term in the tail bound of Lemma 6 is the linear term ct. So Lemma 6 gives that:

$$\Pr\left[\sum_{i:x_i<0}|x_i|^{p-1} < (k-k')\mu - 2\sqrt{2k\mu\log(1/\delta)}\right] \le \delta/6,$$

$$\Pr\left[\sum_{i:x_i\ge 0}|x_i|^{p-1} > k'\mu + 2\sqrt{2k\mu\log(1/\delta)}\right] \le \delta/6.$$

Combined with the Chernoff bound, this gives that with probability $1 - 2\delta/3$:

$$\begin{aligned}
&-\left(1-\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i<0}|x_i|^{p-1}+\left(1+\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i\geq 0}|x_i|^{p-1}\\
\leq&-\left(1-\frac{2p}{\sqrt{k}}\right)\left((k-k')\mu-(2\sqrt{2})\sqrt{k\mu\log(1/\delta)}\right) \qquad (6)\\
&+\left(1+\frac{2p}{\sqrt{k}}\right)\left(k'\mu+(2\sqrt{2})\sqrt{k\mu\log(1/\delta)}\right)\\
\leq&(2k'-k)\mu+(2\sqrt{k}p)\mu+(4\sqrt{2})\sqrt{k\mu\log(1/\delta)}\\
\leq&6\mu\sqrt{k\log(1/\delta)}+(2\sqrt{k}p)\mu+(5\sqrt{2})\mu\sqrt{kp\log(1/\delta)}\\
\leq&16k\mu\cdot\frac{\sqrt{p\log(1/\delta)}}{\sqrt{k}}. \qquad (7)
\end{aligned}$$

In the last step, we use that $p \leq \log k \leq \log(1/\delta)$ for the range of p, δ we consider. On the other hand, by Fact 5 $\sum_{i\in[k]}|x_i|^{p-1} = ||x||_{p-1}^{p-1}$ is sampled from a random variable in $\Gamma^-(k\mu, 3/2)$ and thus by Lemma 13 and Lemma 6 is at least $k\mu/2$ with probability at least $1-\delta/3$, i.e. $k\mu \leq 2||x||_{p-1}^{p-1}$ with probability at least $1-\delta/3$. Combined with (7) by a union bound we get with probability $1-\delta$:

$$-\left(1-\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i<0}|x_i|^{p-1}+\left(1+\frac{2p}{\sqrt{k}}\right)\sum_{i:x_i\geq 0}|x_i|^{p-1}\leq 32\frac{\sqrt{p\log(1/\delta)}}{\sqrt{k}}\cdot||x||_{p-1}^{p-1}$$

Finally, by the Cauchy-Schwarz inequality for any $a \leq b$ and k-dimensional x we have $||x||_a \leq k^{1/a-1/b}||x||_b$. So, $||x||_{p-1}^{p-1} \leq k^{1/p}||x||_p^{p-1}$, giving (5) with probability $1-\delta$ as desired. ◀

Given Lemma 11, determining the value of σ that makes $\mathcal{M}_\sigma^p$ private is fairly straightforward:

▶ **Lemma 14.** *Let $\mathcal{M}_\sigma^p$ be the mechanism such that $\mathcal{M}_\sigma^p(d)$ samples $x \in \mathbb{R}^k$ from $x \sim$ GGauss(p,σ) and outputs $\tilde{d} = d + x$. For $4 \leq p \leq \log k$ that is an even integer, $\epsilon \leq O(1)$, $\delta \in [2^{-O(k/p)}, 1/k]$, and*

$$\sigma = \Theta\left(\frac{\sqrt{kp\log(1/\delta)}}{\epsilon}\right),$$

$\mathcal{M}_\sigma^p$ is (ϵ,δ)-differentially private.

Proof. It suffices to show that for any vector Δ in $[-1,1]^k$:

$$\Pr_{\tilde{d}\sim\mathcal{M}_\sigma^p(d)}\left[\log\left(\frac{\Pr[\mathcal{M}_\sigma^p(d)=\tilde{d}]}{\Pr[\mathcal{M}_\sigma^p(d+\Delta)=\tilde{d}]}\right)\leq\epsilon\right]=\Pr_{\tilde{d}\sim\mathcal{M}_\sigma^p(d)}\left[\frac{||x-\Delta||_p^p-||x||_p^p}{\sigma^p}\leq\epsilon\right]\geq 1-\delta.$$

Here, we abuse notation by letting Pr also denote a likelihood function. By Lemma 11 we now have with probability $1-\delta/2$ for a sufficiently large constant c:

$$||x-\Delta||_p^p-||x||_p^p\leq 64pk^{1/p-1/2}\sqrt{p\log(1/\delta)}||x||_p^{p-1}+2p^2k^{\frac{p}{2}}.$$

The pdf of the rescaled norm $r = ||x||_p/\sigma$ is proportional to $r^{k-1}\exp(-r^p)$ over $(0,\infty)$ (the r^{k-1} appears because the $(k-1)$-dimensional surface area of the ℓ_p-sphere of radius r is proportional to r^{k-1}). Letting R denote r^p, the pdf of R is proportional to $R^{\frac{k}{p}-1}\exp(-R)$

by change of variables, i.e. R is the random variable $Gamma(\frac{k}{p})$. Then by the Gamma tail bound, with probability at least $1 - e^{-.001k/p} > 1 - \delta/2$, R is contained in $[\frac{k}{2p}, \frac{2k}{p}]$, so $||x||_p$ is contained in $[\sigma\left(\frac{k}{2p}\right)^{1/p}, \sigma\left(\frac{2k}{p}\right)^{1/p}]$. Then by a union bound, with probability $1 - \delta$:

$$\frac{||x - \Delta||_p^p - ||x||_p^p}{\sigma^p} \leq \frac{128p^{1/p}\sqrt{kp\log(1/\delta)}}{\sigma} + \frac{4p^2k^{\frac{p}{2}}}{\sigma^p}.$$

Noting that $n^{1/n}$ is contained within $[1, e^{1/e}]$ for all $n \geq 1$, letting

$$\sigma = 185 \cdot \frac{\sqrt{kp\log(1/\delta)}}{\epsilon},$$

we get that $\frac{||x-\Delta||_p^p-||x||_p^p}{\sigma^p} \leq \epsilon$ with probability $1 - \delta$ as desired. ◀

2.2 Error Guarantees

In this section, we analyze the ℓ_∞ error of $\mathcal{M}_\sigma^p$, for a given choice of δ in the range specified in Lemma 14. We give an expected error bound, and also a tail bound on the error. The error analysis follows almost immediately from the following lemma, which bounds the fraction of a sphere cap's volume with a large first coordinate:

▶ **Lemma 15.** *Let x be chosen uniformly at random from a k-dimensional ℓ_p-sphere with arbitrary radius, i.e. the set of points with $||x||_p = R$ for some R, for $p \geq 1$. Then we have:*

$$\Pr[|x_1| \geq r||x||_p] \leq (1 - r^p)^{(k-1)/p} \leq \exp\left(-\frac{(k-1)r^p}{p}\right).$$

This lemma or one providing a similar bound likely already exists in the literature, but we are unaware of a reference for it. So, for completeness we give the full proof at the end of the section.

▶ **Corollary 16.** *Let x be chosen uniformly at random from a k-dimensional ℓ_p-sphere with arbitrary radius for $p \geq 1$. Then we have:*

$$\Pr[||x||_\infty \geq r||x||_p] \leq k \cdot \exp\left(-\frac{(k-1)r^p}{p}\right).$$

Proof. This follows from Lemma 15 and a union bound over all k coordinates (which have identical marginal distributions). ◀

Combining this corollary with Lemma 14, it is fairly straightforward to prove our first main result:

▶ **Theorem 17.** *Let $\mathcal{M}_\sigma^p$ be the mechanism such that $\mathcal{M}_\sigma^p(d)$ samples $x \in \mathbb{R}^k$ from $GGauss(p, \sigma)$ and outputs $\tilde{d} = d + x$. For $4 \leq p \leq \log k$ that is an even integer, For $\epsilon \leq O(1)$, $\delta \in [2^{-O(k/p)}, 1/k]$, and*

$$\sigma = 185 \cdot \frac{\sqrt{kp\log(1/\delta)}}{\epsilon},$$

$\mathcal{M}_\sigma^p$ is (ϵ, δ)-differentially private and for some sufficiently large constant c, and all $t \geq 0$:

$$\Pr_{\tilde{d}\sim\mathcal{M}_\sigma^p(d)}\left[||\tilde{d} - d||_\infty \geq 1480t \cdot \frac{\sqrt{kp}\log^{1/p}k\sqrt{\log(1/\delta)}}{\epsilon}\right] \leq e^{-t^p\log k} + e^{-.001k/p}$$

Proof. The privacy guarantee follows from Lemma 14.

For the tail bound, if $||\tilde{d} - d||_\infty > 1480t \cdot \frac{\sqrt{k}\log^{1/p} k\sqrt{p\log(1/\delta)}}{\epsilon}$ we have either $||x||_p \geq 370 \cdot \frac{k^{1/2+1/p}\sqrt{p\log(1/\delta)}}{\epsilon}$ or $||x||_\infty > \frac{4t\log^{1/p} k}{k^{1/p}}||x||_p$. Recall that $(||x||_p/\sigma)^p$ is distributed according to a $Gamma(\frac{k}{p})$ random variable, and thus by a Gamma tail bound exceeds $2k/p$ with probability at most $e^{-.001k/p}$. In turn, $||x||_p \geq 370 \cdot \frac{k^{1/2+1/p}\sqrt{p\log(1/\delta)}}{\epsilon} \geq \left(\frac{2k}{p}\right)^{1/p}\sigma$ with at most this probability. Then it follows by setting $r = \frac{4t\log^{1/p} k}{k^{1/p}}$ in Corollary 16 and a union bound that:

$$\Pr\left[||\tilde{d} - d||_\infty \geq 1480t \cdot \frac{\sqrt{k}\log^{1/p} k\sqrt{p\log(1/\delta)}}{\epsilon}\right] \leq \Pr\left[||x||_\infty \geq \frac{4t\log^{1/p} k}{k^{1/p}}||x||_p\right]$$

$$+e^{-.001k/p} \leq \exp\left(-\frac{(k-1)4^p t^p \log k}{kp}\right) + e^{-.001k/p} \leq e^{-t^p \log k} + e^{-.001k/p}. \quad ◀$$

This proves Theorem 1, up to some details which we defer to Section A.

2.3 Proof of Lemma 15

To prove this lemma we'll need the following lemma about convex bodies.

▶ **Lemma 18.** *Let $A \subseteq B \subset \mathbb{R}^k$ be two compact convex bodies with A contained in B, and A', B' be their respective boundaries. Then $Vol_{k-1}(A') \leq Vol_{k-1}(B')$, where Vol_{k-1} denotes the $(k-1)$-dimensional volume.*

Proof. For any compact convex body S and its boundary S', the $(k-1)$-dimensional volume of S' satisfies:

$$\mathrm{Vol}_{k-1}(S') \propto \int_{\mathbb{S}^k} \mathrm{Vol}_{k-1}(\pi_{\theta^\top} S)\mathrm{d}\theta,$$

Where $\mathbb{S}^k$ is the k-dimensional unit sphere and $\pi_{\theta^\top} S$ is the orthogonal projection of S onto the subspace of $\mathbb{R}^k$ orthogonal to θ (see e.g. Section 5.5 of [13] for a proof of this fact). Since $A \subseteq B$ it follows that for all θ we have $\mathrm{Vol}_{k-1}(\pi_{\theta^\top} A) \leq \mathrm{Vol}_{k-1}(\pi_{\theta^\top} B)$ and so $\mathrm{Vol}_{k-1}(A') \leq \mathrm{Vol}_{k-1}(B')$. ◀

The idea behind the proof of Lemma 15 is to show that the region of the ℓ_p-ball with large positive first coordinate is contained within a smaller ℓ_p-ball, and then apply Lemma 18:

Proof of Lemma 15. By rescaling, we can assume $||x||_p = 1$ and instead show:

$$\Pr[|x_1| \geq r] \leq (1 - r^p)^{(k-1)/p}$$

$$\Pr[|x_1| \geq r] = \frac{\mathrm{Vol}_{k-1}\left(\{x : |x_1| \geq r, ||x||_p = 1\}\right)}{\mathrm{Vol}_{k-1}\left(x : ||x||_p = 1\right)} = \frac{\mathrm{Vol}_{k-1}\left(\{x : x_1 \geq r, ||x||_p = 1\}\right)}{\mathrm{Vol}_{k-1}\left(\{x : x_1 \geq 0, ||x||_p = 1\}\right)},$$

Where Vol_{k-1} denotes the $(k-1)$-dimensional volume. To bound this ratio, let v be the vector $(r, 0, 0, \ldots, 0)$, and consider the (compact, convex) body $B_1 = \{x : x_1 \geq r, ||x - v||_p \leq (1 - r^p)^{1/p}\}$. We have $r^p + (v - r)^p \leq v^p$ for $0 \leq r \leq v$, so B_1 contains the (also compact, convex) body $B_2 = \{x : x_1 \geq r, ||x||_p \leq 1\}$. Then by Lemma 18 the $(k-1)$-dimensional surface area of B_1 is larger than that of B_2. The boundary of B_1 is the union of the bodies $B_{1,a} := \{x : x_1 = r, ||x - v||_p \leq (1 - r^p)^{1/p}\}$ and $B_{1,b} := \{x : x_1 \geq r, ||x - v||_p = (1 - r^p)^{1/p}\}$, whose intersection has $(k-1)$-dimensional volume 0. Similarly, the boundary of B_2 is the

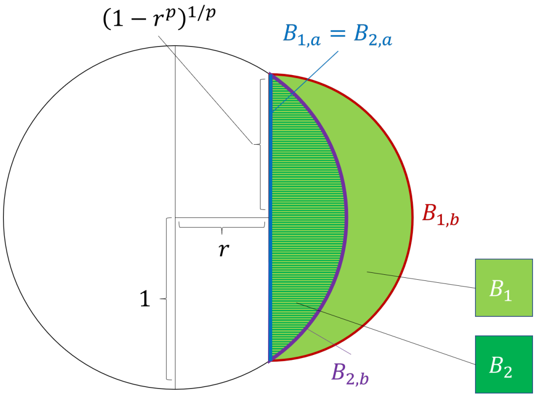


Figure 1 A picture of the bodies in the proof of Lemma 15 for $p = 2, k = 2$. B_2 has stripes that are the same color as $B_1 \setminus B_2$ to emphasize that B_1 contains B_2.

union of the bodies $B_{2,a} := \{x : x_1 = r, ||x||_p \leq 1\}$ and $B_{2,b} := \{x : x_1 \geq r, ||x||_p = 1\}$, whose intersection has $(k-1)$-dimensional volume 0. See Figure 1 for an example of a picture of all of these bodies.

Nothing that $B_{1,a} = B_{2,a}$, we conclude that $\text{Vol}_{k-1}(B_{1,b}) \geq \text{Vol}_{k-1}(B_{2,b})$. Now we have:

$$\frac{\text{Vol}_{k-1}\left(\{x : x_1 \geq r, ||x||_p = 1\}\right)}{\text{Vol}_{k-1}\left(\{x : x_1 \geq 0, ||x||_p = 1\}\right)} \leq \frac{\text{Vol}_{k-1}(\{x : x_1 \geq r, ||x-v||_p = (1-r^p)^{1/p}\})}{\text{Vol}_{k-1}\left(\{x : x_1 \geq 0, ||x||_p = 1\}\right)}.$$

The body in the numerator of the final expression is the body in the denominator, but shifted by v and rescaled by $(1-r^p)^{1/p}$ in every dimension. So, the final ratio is at most $(1-r^p)^{(k-1)/p}$. ◀

3 Composition with Sparse Vector

In this section, we generalize the mechanism of [18], which is a composition of the Gaussian mechanism and sparse vector mechanism of [7], by analyzing a composition of $\mathcal{M}_\sigma^p$ and the sparse vector mechanism instead[3]. The guarantees given by sparse vector can be given in the following form that we will use:

▶ **Theorem 19** (Sparse Vector). *For every $k \geq 1, c_{SV} \leq k, \epsilon_{SV}, \delta_{SV}, \beta_{SV} > 0$, and*

$$\alpha_{SV} \geq O\left(\frac{\sqrt{c_{SV}\log(1/\delta_{SV})}\log(k/\beta_{SV})}{\epsilon_{SV}}\right),$$

there exists a mechanism SV that takes as input $d \in \mathbb{R}^k$ and outputs $\tilde{d} \in \mathbb{R}^k$ such that:

- *SV is $(\epsilon_{SV}, \delta_{SV})$-differentially private.*
- *If at most c_{SV} entries of d have absolute value strictly greater than $\alpha_{SV}/2$, then:*

$$\Pr_{\tilde{d} \sim SV(d)}\left[||\tilde{d} - d||_\infty \geq \alpha_{SV}\right] \leq \beta_{SV}.$$

- *Regardless of the value of d we have for all $t \geq 0$:*

$$\Pr_{\tilde{d} \sim SV(d)}[||\tilde{d} - d|| \geq \max\{||d||_\infty, t\sqrt{k\log(1/\delta_{SV})}/\epsilon_{SV})] \leq ke^{-\Omega(t)}.$$

[3] Unlike its preprint, the journal version of [18] uses a slightly different mechanism based on the exponential mechanism in place of the sparse vector mechanism. A similar change can likely be made to the mechanism given in this section; we stick to using the sparse vector mechanism for a slightly simpler proof.

The proof is deferred to Section A. We now prove Theorem 20, from which Theorem 2 follows up to some minor details (see Section A):

▶ **Theorem 20.** *For any $4 \leq p \leq \log k$ that is an even integer, $\epsilon \leq O(1)$, $\delta \in [2^{-O(k/p)}, 1/k]$, and $t \in [0, O(\frac{\log k}{\log \log k})]$, there exists a (ϵ, δ)-differentially private mechanism $\mathcal{M}$ that takes in a vector $d \in \mathbb{R}^k$ and outputs a random $\tilde{d} \in \mathbb{R}^k$ such that for a sufficiently large constant c :*

$$\Pr_{\tilde{d} \sim \mathcal{M}(d)} \left[||\tilde{d} - d||_\infty \geq \frac{ct\sqrt{kp\log(1/\delta)}(\log\log k)^{1/p}}{\epsilon} \right] \leq e^{-\log^t k}.$$

Proof. The mechanism is as follows: We sample $x \sim GGauss(p, \sigma)$ for

$$\sigma = \Theta\left(\frac{\sqrt{kp\log(1/\delta)}}{\epsilon}\right),$$

If $||x||_p^p > 2k\sigma^p/p$, we output d. Otherwise, we instantiate SV from Theorem 19 with parameters:

$$\alpha_{SV} = 12t(\log\log k)^{1/p}\sigma \leq \frac{ct\sqrt{kp\log(1/\delta)}(\log\log k)^{1/p}}{\epsilon}, \qquad c_{SV} = 4k/\log^{2+2t} k,$$

$$\epsilon_{SV} = \epsilon/2, \qquad \delta_{SV} = \delta/3, \qquad \beta_{SV} = \exp(-\log^t k)/2.$$

We input x to SV to sample $\hat{x}$, and then output $\tilde{d} = d + x - \hat{x}$.

First, note that:

$$\frac{\sqrt{c_{SV}\log(1/\delta_{SV})}\log(k/\beta_{SV})}{\epsilon_{SV}} \leq \frac{\sqrt{\frac{16k}{\log^{2+2t} k}\log(1/\delta)}(\log k + \log^t k)}{\epsilon} \leq \frac{4\sqrt{k\log(1/\delta)}}{\epsilon},$$

i.e. α satisfies the requirements of Theorem 19 as long as the constant hidden in the $\Theta(\cdot)$ notation in the choice of σ is sufficiently large.

To analyze the privacy guarantee, this is the composition of:

- The mechanism of Theorem 17, which if the constant hidden in the $\Theta(\cdot)$ in the expression for σ is sufficiently large, is $(\epsilon/2, \delta/3)$-differentially private.
- The SV mechanism of Theorem 19, with parameters set so it is $(\epsilon/2, \delta/3)$-differentially private.
- The event that $||x||_p^p > 2k\sigma^p/p$, causing us to release the database, which we recall from the Proof of Theorem 17 happens with probability at most $2^{-\Omega(k/p)} \leq \delta/3$.

By composition, we get that the mechanism is (ϵ, δ)-differentially private as desired.

To show the tail bound on ℓ_∞-error: If $||x||_p^p > 2k\sigma^p/p$, then we have $\tilde{d} = d$, so trivially the tail bound is satisfied. So, it suffices to show that conditional on $||x||_p^p \leq 2k\sigma^p/p$ occurring, we have the tail bound. By a union bound, the guarantees of Theorem 19 give that $||\tilde{d} - d||_\infty = ||x - \hat{x}||_\infty \leq \alpha_{SV}$ (i.e the tail bound is satisfied) if at most $4k/\log^{2+2t} k$ entries of x have absolute value greater than $\alpha_{SV}/2$ with probability less than, say, $e^{-2\log^t k}$. Using $r = 3t\frac{(\log\log k)^{1/p}}{k^{1/p}}$ in Lemma 15 and a union bound with the $1 - \delta/3$ probability event that $||x||_p \leq (2k/p)^{1/p}\sigma$, for each coordinate x_i of x we have:

$$|x_i| \geq \alpha_{SV}/2 = 6t(\log\log k)^{1/p}\sigma = 2rk^{1/p}\sigma \geq r||x||_p,$$

with probability at most $\frac{1}{\log^{2+2t} k} + 2^{-\Omega(k/p)} \leq \frac{2}{\log^{2+2t} k}$. Since we sample x with probability proportional to $\exp(-\sum_{i \in [k]} |x_i|^p/\sigma^p)$, each coordinate's distribution is independent, so using a Chernoff bound we conclude that with probability $e^{-\Omega(k/\log^{2+2t} k)} \leq e^{-2\log^t k}$ at most $4k/\log^{2+2t} k$ coordinates have absolute value greater than α_{SV} as desired. ◀

4 Future Directions

As mentioned before, we did not attempt to optimize the constant multiplier in Theorem 1, and our resulting constant is likely too large to be practical. Since the Generalized Gaussian generalizes the Laplace and Gaussian mechanisms, which have good multiplicative constants in practice, we expect that a more careful analysis of the Generalized Gaussian will also lead to a error bound that is practical.

Another question concerns stronger measures of privacy than (ϵ, δ)-DP, including Rényi-DP [16] and zero-concentrated-DP [2]. To show the Generalized Gaussian mechanism satisfies these notions of privacy requires one to bound a moment generating function of the privacy loss $\frac{||x-\Delta||_p^p - ||x||_p^p}{\sigma^p}$, which in some sense requires the privacy loss to be subexponential. Roughly speaking, our analysis shows with probability at least $1-\delta$, the privacy loss lies in an interval in which it behaves as a subgaussian random variable. However, past this interval, our analysis fails to show it even behaves subexponentially. This is because our use of the gamma tail bound of Lemma 6 weakens at two points in the regime where $\delta < 2^{-k/p}$. The first is that the final expression in (7) has a dependence on δ of $\log(1/\delta)$ instead of $\sqrt{\log(1/\delta)}$ when $\delta < 2^{-k/p}$, since the linear term ct in Lemma 6 begins to dominate the error. The second is that, roughly speaking, we use the gamma tail bound to show that $||x||_p^p$ deviates from its expectation of k/p by at most $\sqrt{k\log(1/\delta)/p}$ with probability $1-\delta$. When $\delta \geq 2^{-k/p}$, this lets us treat $||x||_p^p$ as always being within a constant factor of its expectation in our analysis. However, when δ is small enough, the term $\sqrt{k\log(1/\delta)/p}$ becomes much larger than the term k/p, and so we can only bound $||x||_p^p$'s deviation from its expectation by an expression with $\sqrt{\log(1/\delta)}$ dependence on δ.

Our final tail bound on the privacy loss is effectively a product of the tail bound of Lemma 11 and the tail bound on $||x||_p^{p-1}$, and so it shows concentration that is worse than sub-exponential in the small δ regime, which is insufficient for proving these stronger notions of privacy. We believe this is a function of our analysis rather than of the Generalized Gaussian mechanism, but do not know of an alternate analysis that confirms this belief. Determining whether Generalized Gaussian mechanisms can satisfy stronger notions of privacy for larger values of p is an interesting open direction.

References

1 S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford, 2013. URL: `https://books.google.com/books?id=koNqWRluhP0C`.

2 Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 635–658, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

3 Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 1–10, New York, NY, USA, 2014. Association for Computing Machinery. `doi:10.1145/2591796.2591877`.

4 Yuval Dagan and Gil Kur. A bounded-noise mechanism for differential privacy, 2020. `arXiv:2012.03817`.

5 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

6 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

7 Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 381–390, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536467.

8 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014. doi:10.1561/0400000042.

9 Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On avoiding the union bound when answering multiple differentially private queries, 2020. arXiv:2012.09116.

10 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, October 2010. doi:10.1109/FOCS.2010.85.

11 Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 705–714, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806786.

12 N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions.* John Wiley & Sons Incorporated, 1995. URL: https://books.google.com/books?id=q03oAAAACAAJ.

13 D.A. Klain, G.C. Rota, and L.A.R. di Brozolo. *Introduction to Geometric Probability.* Lezioni Lincee. Cambridge University Press, 1997. URL: https://books.google.com/books?id=Q1ytkNM6BtAC.

14 Fang Liu. Generalized gaussian mechanism for differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 31:747–756, 2019.

15 Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, page 94–103, USA, 2007. IEEE Computer Society. doi:10.1109/FOCS.2007.41.

16 Ilya Mironov. Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, 2017.

17 Saralees Nadarajah. A generalized normal distribution. *Journal of Applied Statistics*, 32(7):685–694, 2005. doi:10.1080/02664760500079464.

18 Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *Journal of Privacy and Confidentiality*, 7(2), 2017. doi:10.29012/jpc.v7i2.648.

A Deferred Proofs

A.1 Proof of Theorem 19

Proof of Theorem 19. The mechanism is given by modifying the NumericSparse algorithm given as Algorithm 3 in [8] by outputting 0 instead of $\perp$ or 0 for all remaining queries instead of halting prematurely. The first two properties follow from the associated proofs in that text.

The third property follows because for all entries of $\tilde{d}$ that SV does not output as 0 (for which the error, i.e. corresponding entry of $\tilde{d} - d$, is of course bounded by $||d||_\infty$), the error is drawn from $Lap(b)$ where $b = O(\sqrt{k \log(1/\delta_{SV})}/\epsilon_{SV})$. So the maximum error for these (at most $c_{SV} \leq k$) entries is stochastically dominated by the maximum of the absolute value of k of these Laplace random variables, which is at most tb with probability ke^{-t}. ◀

A.2 Proof of Theorem 1

We first need the following corollary of Lemma 15:

▶ **Corollary 21.** *Let x be chosen uniformly at random from a k-dimensional ℓ_p-sphere with arbitrary radius for $p \geq 1$. Then we have:*

$$\mathbb{E}[||x||_\infty] \leq \frac{5\log^{1/p} k}{k^{1/p}}||x||_p$$

Proof. Since $||x||_\infty/||x||_p$ takes values in $[0,1]$, by Lemma 15 we have:

$$\begin{aligned}\mathbb{E}[||x||_\infty/||x||_p] &= \int_0^1 \Pr[||x||_\infty/||x||_p \geq r]\mathrm{d}r \\ &\leq \int_0^{\frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}}} 1\mathrm{d}r + \int_{\frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}}}^1 k\cdot\exp\left(-\frac{(k-1)r^p}{p}\right)\mathrm{d}r \\ &\leq \frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}} + \int_{\frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}}}^1 k\cdot\exp\left(-\frac{(k-1)2^{p+1}\log k}{kp}\right)\mathrm{d}r \\ &\leq \frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}} + \int_{\frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}}}^1 k\cdot\exp\left(-2\log k\right)\mathrm{d}r \\ &\leq \frac{2^{1+1/p}\log^{1/p}k}{k^{1/p}} + \frac{1}{k} \\ &\leq \frac{5\log^{1/p}k}{k^{1/p}}.\end{aligned}$$

Here we use that $2^p \geq p$ for all $p \geq 1$ and that $(1-\frac{c}{x})^x \leq e^{-c}$ for all $c \geq 0$. ◀

Proof of Theorem 1. We use Theorem 17 after rounding p up to the nearest even integer (this loses at most a multiplicative constant in the resulting error bounds). If the constant hidden in $\Theta(\log\log k)$ is a sufficiently large function of c_1, this gives the desired tail bound, up to the additive $e^{-.001k/p}$ in the probability bound (which may be larger than the $e^{-t^p \log k}$ term for large values of p). To remove the additive $e^{-.001k/p}$: if the less than $e^{-.001k/p} \leq \delta$ probability event that $(||x||_p/\sigma)^p$ exceeds $2k/p$ occurs, we can instead just output $\tilde{d} = d$, i.e. instead set $x = 0$. This gives an $(\epsilon, 2\delta)$-private mechanism that always satisfies $(||x||_p/\sigma)^p \leq 2k/p$, and then we can rescale our choice of δ appropriately. The tail bound can now be derived as in the proof of Theorem 17. Similarly, since we always have $(||x||_p/\sigma)^p \leq 2k/p$, the expectation of $||x||_\infty$ follows from Corollary 21. Finally, the expectation of $||x||_q$ for $1 \leq q \leq p$ follows by using Jensen's inequality twice and the unconditional upper bound on $||x||_p^p$:

$$\mathbb{E}[||x||_q] \leq \mathbb{E}[||x||_q^q]^{1/q} = k^{1/q}\mathbb{E}[|x_1|^q]^{1/q} \leq k^{1/q}\mathbb{E}[|x_1|^p]^{1/p} = k^{1/q-1/p}\mathbb{E}[||x||_p^p]$$

$$\leq k^{1/q-1/p}\cdot(2k/p)^{1/p}\sigma = O(k^{1/q}\sigma).$$ ◀

A.3 Proof of Theorem 2

Proof of Theorem 2. The tail bound in Theorem 2 follows immediately from Theorem 20 by choosing p to be an even integer satisfying $p = \Theta(\log\log\log k)$.

For the expectation, we use the tail bound of Theorem 2. We have:

$$\mathbb{E}_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty\right] = \int_0^\infty \Pr[||\tilde{d}-d||_\infty \geq s]\mathrm{d}s$$

$$= \int_0^a \Pr[||\tilde{d}-d||_\infty \geq s]\mathrm{d}s + \int_a^b \Pr[||\tilde{d}-d||_\infty \geq s]\mathrm{d}s + \int_b^\infty \Pr[||\tilde{d}-d||_\infty \geq s]\mathrm{d}s.$$

We choose $a = \frac{2c\sqrt{k\log\log\log k\log(1/\delta)}}{\epsilon}$, $b = \frac{k\sqrt{\log(1/\delta)}}{\epsilon}$. The integral over $[0, a]$ is of course bounded by a. By Theorem 20, the integral over $[a, b]$ is bounded by $b \cdot e^{-\log^2 k} \leq \frac{\sqrt{\log(1/\delta)}}{\epsilon} \leq a$. Finally, to bound the third term, recall that the mechanism of Theorem 20 outputs d (i.e. effectively chooses $x, \hat{x} = 0$ instead) if $||x||_p$ is too large. So, unconditionally we have:

$$||x||_\infty \leq ||x||_p \leq (2k/p)^{1/p}\sigma \leq \frac{2c\sqrt{k\log\log\log k\log(1/\delta)}}{\epsilon} \leq b.$$

So by the third property in Theorem 19 we have for $s \in [b, \infty)$:

$$\Pr_{\tilde{d}\sim\mathcal{M}(d)}[||\tilde{d}-d||_\infty \geq s] = \Pr_{x,\hat{x}}[||x-\hat{x}||_\infty \geq s] \leq ke^{-\Omega(s/(\sqrt{k\log(1/\delta)}/\epsilon))}.$$

And so by change of variables, with $s' = s/(\sqrt{k\log(1/\delta)}/\epsilon)$:

$$\int_b^\infty \Pr[||\tilde{d}-d||_\infty \geq s]\mathrm{d}s \leq \frac{\sqrt{k\log(1/\delta)}}{\epsilon}\int_{\sqrt{k}}^\infty ke^{-\Omega(s')}\mathrm{d}s' \leq \frac{k^{1.5}\sqrt{\log(1/\delta)}}{\epsilon} \cdot e^{-\Omega(\sqrt{k})} \leq a.$$

So we conclude

$$\mathbb{E}_{\tilde{d}\sim\mathcal{M}(d)}\left[||\tilde{d}-d||_\infty\right] \leq 3a = O\left(\frac{\sqrt{k\log\log\log k\log(1/\delta)}}{\epsilon}\right),$$

as desired. ◀

B Concentration of Generalized Gammas

In this section we consider the Generalized Gamma random variable $GGamma(a, b)$ parameterized by a, b with pdf:

$$p(x) = \frac{bx^{a-1}e^{-x^b}}{\Gamma(a/b)}, x \in (0, \infty).$$

Where the Gamma function $\Gamma(x)$ is defined over the positive reals as

$$\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}\mathrm{d}x.$$

We recall that $\Gamma(z)$ is a continuous analog of the factorial in that it satisfies $\Gamma(x+1) = x \cdot \Gamma(x)$. When $b = 1$, $GGamma(a, b)$ is exactly the Gamma random variable $Gamma(a)$ (we will use $Gamma$ to denote the random variable and Γ to denote the function to avoid ambiguous notation).

We want to show that sums of $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ random variables concentrate nicely. To do this, we will show that they are sub-gamma:

To show that $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ are sub-gamma, we will relate the moment-generating function of $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ to that of the Gamma random variable with the same mean using the following facts:

▶ **Fact 22.** *For a Generalized Gamma random variable $X \sim GGamma(a, b)$ the moments are $\mathbb{E}[X^r] = \frac{\Gamma((a+r)/b)}{\Gamma(a/b)}$. In particular, for a Gamma random variable $X \sim Gamma(a)$ the moments are $\mathbb{E}[X^r] = \frac{\Gamma(a+r)}{\Gamma(a)}$.*

See e.g. Section 17.8.7 of [12] for a derivation of this fact. Note here that $GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ has mean $\mu = 1/\Gamma(1/p)$. To relate the moments of Generalized Gamma random variables to Gamma random variables' we note the following about μ:

▶ **Fact 23.** *For all $p \geq 2$, we have $\frac{1}{p} \leq \frac{1}{\Gamma(1/p)} \leq \frac{1.2}{p}$.*

Putting it all together, we get the following lemmas, which combined with Fact 23 give us Lemma 13:

▶ **Lemma 24.** *Let $Y = GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ for $p \geq 2$. Then, for $\mu = \mathbb{E}[Y] = \frac{1}{\Gamma(1/p)}$, we have $Y \in \Gamma^+(\mu, 1)$.*

Proof. We compare the moment-generating function of (the centered version of) Y to that of $X = Gamma(\mu)$ where $\mu = \mathbb{E}[Y]$. X is in $\Gamma(\mu, 1)$ so it suffices to show Y's moment generating function is smaller than X's. First, looking at the moment generating function of Y, we have:

$$\begin{aligned}\mathbb{E}[e^{\lambda Y}] &= 1 + \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{\lambda^r}{r!}\mathbb{E}[Y^r]\right] \\ &= 1 + \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{\lambda^r}{r!}\frac{\Gamma(\frac{1}{p} + \frac{r(p-1)}{p})}{\Gamma(\frac{1}{p})}\right] \\ &\overset{(a)}{\leq} 1 + \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{\lambda^r}{r!}\frac{\Gamma(\frac{1}{p} + r)}{\Gamma(\frac{1}{p})}\right] \\ &\overset{(b)}{\leq} 1 + \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{\lambda^r}{r!}\frac{\Gamma(\mu + r)}{\Gamma(\mu)}\right] = \mathbb{E}[e^{\lambda X}].\end{aligned}$$

(a) follows because the Gamma function is monotonically increasing in the range $[1.5, \infty)$. (b) follows because $\mu = \frac{1}{\Gamma(1/p)} \geq 1/p$ for $p \geq 1$, and because for positive integers r, $\frac{\Gamma(x+r)}{\Gamma(x)} = \prod_{i=0}^{r-1}(x+i)$ is monotonically increasing in x. Since $X \in \Gamma^+(\mu, 1)$ and X, Y have the same mean, we have that $Y \in \Gamma^+(\mu, 1)$ as well. ◀

▶ **Lemma 25.** *Let $Y = GGamma(\frac{1}{p-1}, \frac{p}{p-1})$ for $p \geq 3$. Then, for $\mu = \mathbb{E}[Y] = \frac{1}{\Gamma(1/p)}$, we have $Y \in \Gamma^-(\mu, 3/2)$.*

Proof. Similarly to the previous lemma, we have for all $0 \leq \lambda \leq 2/3$:

$$\begin{aligned}&\mathbb{E}[e^{-\lambda Y}] \\ &= 1 - \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{(-\lambda)^r}{r!}\frac{\Gamma(\frac{1}{p} + \frac{r(p-1)}{p})}{\Gamma(\frac{1}{p})}\right] \\ &= 1 - \lambda\mu + \sum_{r=1}^{\infty}\left[\frac{\lambda^{2r}}{(2r)!}\cdot\frac{\Gamma(\frac{1}{p} + 2r\frac{p-1}{p})}{\Gamma(\frac{1}{p})}\left(1 - \frac{\lambda}{2r+1}\cdot\frac{\Gamma(\frac{1}{p} + (2r+1)\frac{p-1}{p})}{\Gamma(\frac{1}{p} + 2r\frac{p-1}{p})}\right)\right] \\ &= 1 - \lambda\mu + \sum_{r=1}^{\infty}\left[\frac{\lambda^{2r}}{(2r)!}\cdot\frac{\Gamma(\frac{1}{p} + 2r)}{\Gamma(\frac{1}{p})}\left(\frac{\Gamma(\frac{1}{p} + 2r\frac{p-1}{p})}{\Gamma(\frac{1}{p} + 2r)} - \frac{\lambda}{2r+1}\cdot\frac{\Gamma(\frac{1}{p} + (2r+1)\frac{p-1}{p})}{\Gamma(\frac{1}{p} + 2r)}\right)\right]\end{aligned}$$

$$\begin{aligned}
\ldots &\overset{(c)}{\leq} 1 - \lambda\mu + \sum_{r=1}^{\infty}\left[\frac{\lambda^{2r}}{(2r)!}\cdot\frac{\Gamma(\frac{1}{p}+2r)}{\Gamma(\frac{1}{p})}\left(1-\frac{\lambda}{2r+1}\cdot\frac{\Gamma(\frac{1}{p}+2r+1)}{\Gamma(\frac{1}{p}+2r)}\right)\right]\\
&\overset{(d)}{\leq} 1 - \lambda\mu + \sum_{r=1}^{\infty}\left[\frac{\lambda^{2r}}{(2r)!}\cdot\frac{\Gamma(\mu+2r)}{\Gamma(\mu)}\left(1-\frac{\lambda}{2r+1}\cdot\frac{\Gamma(\mu+2r+1)}{\Gamma(\mu+2r)}\right)\right]\\
&= 1 - \lambda\mu + \sum_{r=2}^{\infty}\left[\frac{(-\lambda)^r}{r!}\cdot\frac{\Gamma(\mu+r)}{\Gamma(\mu)}\right] = \mathbb{E}[e^{-\lambda X}].
\end{aligned}$$

Which, up to proving $(c), (d)$ hold, shows that $Y \in \Gamma^-(\mu, 3/2)$ since X and Y have the same mean and $X \in \Gamma^-(\mu, 0) \subset \Gamma^-(\mu, 3/2)$. (c) follows because the change in each term in the sum is

$$\frac{\lambda^{2r}}{(2r)!}\frac{1}{\Gamma\left(\frac{1}{p}\right)}\cdot$$

$$\left[\Gamma\left(\frac{1}{p}+2r\right)-\Gamma\left(\frac{1}{p}+2r\frac{p-1}{p}\right)-\frac{\lambda}{2r+1}\left(\Gamma\left(\frac{1}{p}+2r+1\right)-\Gamma\left(\frac{1}{p}+(2r+1)\frac{p-1}{p}\right)\right)\right].$$

To show this expression is non-negative, it suffices to show that just the term in the brackets is positive, or equivalently, for all $r \geq 2, p \geq 3$:

$$\Gamma\left(\frac{1}{p}+2r\right)\left(1-\frac{\Gamma\left(\frac{1}{p}+2r\frac{(p-1)}{p}\right)}{\Gamma\left(\frac{1}{p}+2r\right)}\right) \geq \frac{\lambda}{2r+1}\Gamma\left(\frac{1}{p}+2r+1\right)\left(1-\frac{\Gamma\left(\frac{1}{p}+(2r+1)\frac{p-1}{p}\right)}{\Gamma\left(\frac{1}{p}+2r+1\right)}\right).$$

Since we have $\Gamma\left(\frac{1}{p}+2r+1\right) = (\frac{1}{p}+2r)\Gamma\left(\frac{1}{p}+2r\right) \leq (2r+1)(\frac{1}{p}+2r)$, it further suffices to just show:

$$f(r,p) := \frac{\left(1-\frac{\Gamma(\frac{1}{p}+2r\frac{(p-1)}{p})}{\Gamma(\frac{1}{p}+2r)}\right)}{\left(1-\frac{\Gamma(\frac{1}{p}+(2r+1)\frac{p-1}{p})}{\Gamma(\frac{1}{p}+2r+1)}\right)} \geq \lambda.$$

For any fixed $r \geq 2$, one can verify analytically that $f(r,p)$ is monotonically decreasing in p over $p \in [1, \infty)$ and the limit as p goes to infinity is $g(r) := \frac{2r\psi(2r)}{(2r+1)\psi(2r+1)}$ where ψ is the digamma function $\psi(x) = \frac{\frac{\mathrm{d}}{\mathrm{d}x}\Gamma(x)}{\Gamma(x)}$. One can also verify analytically that $g(r)$ is monotonically increasing, and $g(2) \approx .6672$. So, for all $r \geq 2, p \geq 3$ we have $f(r,p) > 2/3$ and thus for $\lambda \in [0, 2/3]$, the inequality (c) is satisfied.

(d) follows by looking at the function

$$z(x) = \frac{\Gamma(x+r)}{\Gamma(x)}\left(1-\frac{\lambda}{r+1}\cdot\frac{\Gamma(x+r+1)}{\Gamma(x+r)}\right) = \left(1-\frac{\lambda(x+r)}{r+1}\right)\prod_{i=0}^{r-1}(x+i).$$

For $r \geq 2, \lambda \leq 1$, one can verify analytically that $z(x)$ is monotonically increasing in the interval $(0, 1/2] \supseteq (0, \frac{1.2}{p}] \supseteq (0, \mu]$. Since $\mu \geq \frac{1}{p}$, this gives that each term in the right-hand-side of (d) is larger than the corresponding term on the left-hand-side. ◀

An Algorithmic Framework for Fairness Elicitation

Christopher Jung
University of Pennsylvania, Philadelphia, PA, USA

Michael Kearns
University of Pennsylvania, Philadelphia, PA, USA

Seth Neel
Harvard University, Cambridge, MA, USA

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA

Logan Stapleton
University of Minnesota, Minneapolis, MN, USA

Zhiwei Steven Wu
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We consider settings in which the right notion of fairness is not captured by simple mathematical definitions (such as equality of error rates across groups), but might be more complex and nuanced and thus require *elicitation* from individual or collective stakeholders. We introduce a framework in which pairs of individuals can be identified as requiring (approximately) equal treatment under a learned model, or requiring ordered treatment such as “applicant Alice should be at least as likely to receive a loan as applicant Bob”. We provide a provably convergent and *oracle efficient* algorithm for learning the most accurate model subject to the elicited fairness constraints, and prove generalization bounds for both accuracy and fairness. This algorithm can also combine the elicited constraints with traditional statistical fairness notions, thus “correcting” or modifying the latter by the former. We report preliminary findings of a behavioral study of our framework using human-subject fairness constraints elicited on the COMPAS criminal recidivism dataset.

2012 ACM Subject Classification Theory of computation → Machine learning theory

Keywords and phrases Fairness, Fairness Elicitation

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.2

Related Version *Full Version*: `https://arxiv.org/abs/1905.10660`

1 Introduction

The literature on algorithmic fairness has consisted largely of researchers proposing and showing how to impose technical definitions of fairness [18, 32, 66, 3, 4, 34, 60, 48, 29, 26, 4, 14]. Because these imposed notions of fairness are described analytically, they are typically simplistic, and often have the form of equalizing simple error statistics across groups. Our starting point is the observation that:

1. This process cannot result in notions of fairness that do not have any simple, analytic description, and
2. This process also overlooks a more precursory problem: namely, *who gets to define what is fair?*

It’s unlikely that researchers alone are best fit for defining algorithmic fairness. Recent work identifies undue power imbalances [40] and biases [39] that arise when algorithm designers and researchers are the only voices in conversations around ethical design. [59] find that many machine learning practitioners are disconnected from the “organisational

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 2; pp. 2:1–2:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and institutional realities, constraints and needs" specific to the contexts in which their algorithms are applied. Researchers may not be able to propose a concise technical definition, e.g. statistical parity, to capture the nuances of fairness in any given context. Furthermore, many philosophers hold that *stakeholders* who are affected by moral decisions and *experts* who understand the context in which moral decisions are made will have the best judgment about which decisions are fair in that context [63, 39].

To this end, we aim to allow stakeholders and experts to play a central role in the process of defining algorithmic fairness. This is aligned with recent works on *virtual democracy*, which propose and enact participatory methods to automate moral decision-making [13, 51, 31, 46, 21].

The way we involve stakeholders is motivated by two concerns:

1. We want stakeholders to have free rein over how they may define fairness, e.g. we don't want to simply have them vote on whether existing, simple constraints like statistical parity or equalized odds is best; and
2. We want non-technical stakeholders to be able to contribute, even if they may not understand the inner workings of a learning algorithm.

We hold that people often cannot elucidate their conceptions of fairness; yet, they can identify specific scenarios where fairness or unfairness occurs.[1] Drawing from individual notions of fairness like [17, 29] that are defined in terms of pairwise comparisons, we therefore aim to elicit stakeholders conceptions of fairness by asking them to compare pairs of individuals in specific scenarios. Specifically, we ask whether it's fair that one particular individual should receive an outcome that is as desirable or better than the other.

When pointing out fairness or unfairness, this kind of pairwise ranking is natural. For example, after Serena Williams was penalized for a verbal interaction with an umpire in the 2018 U.S. Open Finals, tennis player James Blake tweeted, "I have said worse and not gotten penalized. And I've also been given a "soft warning" by the ump where they tell you knock it off or I will have to give you a violation. [The umpire] should have at least given [Williams] that courtesy" [65]. Here, Blake thinks that: 1) Williams should have been judged as or less severely than he would have been in a similar situation; and 2) the umpire's decision was unfair, because Williams was judged more severely.

Thus, we ask a set of stakeholders about a fixed set of pairs of individuals subject to a classification problem. For each pair of individuals (A, B), we ask the stakeholder to choose from amongst a set of four options:

1. Fair outcomes must classify A and B the *same* way (i.e. they must either both get a favorable classification or both get an unfavorable classification).
2. Fair outcomes must give A an outcome that is equal to *or preferable to* the outcome of B.
3. Fair outcomes must give B an outcome that is equal to *or preferable to* the outcome of A
4. Fair outcomes may treat A and B differently without any constraints.

These constraints, a data distribution, and a hypothesis class define a learning problem: minimize classification error subject to the constraint that the rate of violation of the elicited pairwise constraints is held below some fixed threshold. Crucially and intentionally we elicit relative pairwise orderings of outcomes (e.g. A and B should be treated equally), but do not elicit preferences for absolute outcomes (e.g. A should receive a positive outcome). This is

[1] This is philosophically akin to a theory of moral epistemology called *moral perception*, which claims that we know moral facts (e.g. goodness or fairness) via perception, as opposed to knowing them via rules of morality (see [10]).

because *fairness* – in contrast to *justice* – is often conceptualized as a measure of equality of outcomes, rather than correctness of outcomes[2]. In particular, it remains the job of the learning algorithm to optimize for correctness subject to elicited fairness constraints.

We remark that the premise (and the foundation for the enormous success) of machine learning is that accurate decision making rules in complex scenarios cannot be defined with simple analytic rules, and instead are best derived directly from data. Our work can be viewed similarly, as deriving fairness constraints from data elicited from experts and stakeholders. In this paper, we solve the computational, statistical, and conceptual issues necessary to do this, and demonstrate the effectiveness of our approach via a small behavioral study.

1.1 Results

Our Model

We model individuals as having features in $\mathcal{X}$ and binary labels, drawn from some distribution $\mathcal{P}$. A committee of *stakeholders*[3] $u \in \mathcal{U}$ has preferences about whether one individual should be judged better than another individual. We imagine presenting each stakeholder with a set of pairs of individuals and asking them to choose one of four options for each pair, e.g. given the features of Serena Williams and Jacob Blake:

1. No constraint;
2. Williams should be treated as well as Blake or better;
3. Blake should be treated as well as Williams or better; or
4. Williams and Blake should be treated similarly.

Here, when we refer to how an individual *should be treated*, we mean the probability that an individual is given a positive label by the classifier. This may be a bit of a relaxation of these judgments, since they are not about actualized classifications, but rather the *probabilities* of positive classification. For example, we may not consider it a violation of fairness preference (2) if Williams is judged worse than Blake in a specific scenario; yet, if an ump is *more likely* to judge Williams worse than Blake in general, then this would violate this fairness preference.

We represent these preferences abstractly as a set of ordered pairs $C_u \subseteq \mathcal{X} \times \mathcal{X}$ for each stakeholder u. If $(x, x') \in C_u$, this means that stakeholder u believes that individual x' must be treated as well as individual x or better, i.e. ideally the classifier h classifies such that $h(x') \geq h(x)$. This captures all possible responses above. For example, for Serena Williams (s) and Jacob Blake (b), if stakeholder u responds:

1. *No constraint* $\Leftrightarrow$ $(s, b) \notin C_u$ nor $(b, s) \notin C_u$;
2. *Williams as well as Blake* $\Leftrightarrow$ $(b, s) \in C_u$;
3. *Blake as well as Williams* $\Leftrightarrow$ $(s, b) \in C_u$; or
4. *Treated similarly* $\Leftrightarrow$ $(s, b) \in C_u$ and $(b, s) \in C_u$ (since if $h(b) \geq h(s)$ and $h(s) \geq h(b)$, then $h(s) = h(b)$).

[2] Sidney Morgenbesser, following the Columbia University campus protests in the 1960s, reportedly said that the police had treated him unjustly, but not unfairly. He said that he was treated unjustly because the police hit him without provocation – but not unfairly, because the police were doing the same to everyone else as well.

[3] Though we develop our formalism as a committee of stakeholders, note that it permits the special case of a single subjective stakeholder, which we make use of in our behavioral study.

We impose no structure on how stakeholders form their views nor on the relationship between the views of different stakeholders – i.e. the sets $\{C_u\}_{u \in \mathcal{U}}$ are allowed to be arbitrary (for example, they need not satisfy a triangle inequality), and need not be mutually consistent. We write $C = \cup_u C_u$.

We then formulate an optimization problem constrained by these pairwise fairness constraints. Since it is intractable to require that all constraints in C be satisfied exactly, we formulate two different "knobs" with which we can quantitatively relax our fairness constraints.

For $\gamma > 0$ (our first knob), we say that the classification of an ordered pair of individuals $(x, x') \in C$ satisfies γ-fairness if the probability of positive classification for x' plus γ is no smaller than the probability of positive classification for x, i.e. $\mathbb{E}[h(x')] + \gamma \geq \mathbb{E}[h(x)]$. In this expression, the expectation is taken only over the randomness of the classifier h. Equivalently, a γ-fairness violation corresponds to the classification of an ordered pair of individuals $(x, x') \in C$ if the difference between these probabilities of positive classification is greater than γ, i.e. $\mathbb{E}[h(x) - h(x')] > \gamma$. Thus, γ acts as a buffer on how likely it is that x' be classified worse than x before a fairness violation occurs. For example, if Blake (b) receives a good label (i.e. no penalty) 80% of the time and Williams (s) 50% of the time, then for $\gamma = 0.1$ this constitutes a γ-fairness violation for the ordered pair $(b, s) \in C$, since $\mathbb{E}[h(b) - h(s)] = 0.3 \geq 0.1 = \gamma$.

We might ask that for *no* pair of individuals do we have a γ-fairness violation:

$$\max_{(x,x') \in C} \mathbb{E}[h(x) - h(x')] \leq \gamma.$$

On the other hand, we could ask for the weaker constraint that *over a random draw of a pair of individuals*, the expected fairness violation is at most η (our second knob): $\mathbb{E}_{(x,x') \sim \mathcal{P}^2}[(h(x) - h(x')) \cdot \mathbf{1}[(x, x') \in C]] \leq \eta$. We can also combine both relaxations to ask that the in expectation over random pairs, the "excess" fairness violation, on top of an allowed budget of γ, is at most η. For example, as above, if Blake receives a good label 80% of the time and Williams 50%, for $\gamma = 0.1$, the umpire classifier would pick up 0.2 excess fairness violation for $(b, s) \in C$. In Section 2, we weight these excess fairness violations by the proportion of stakeholders who agree with the corresponding fairness constraint and mandate their sum be less than η. Subject to these constraints, we would like to find the distribution over classifiers that minimizes classification error: given a setting of the parameters γ and η, this defines a benchmark with which we would like to compete.

Our Theoretical Results

Even absent fairness constraints, learning to minimize 0/1 loss (even over linear classifiers) is computationally hard in the worst case (see e.g. [20, 19]). Despite this, learning seems to be empirically tractable in the real world. To capture the *additional* hardness of learning subject to fairness constraints, we follow several recent papers [2, 33] in aiming to develop *oracle efficient* learning algorithms. Oracle efficient algorithms are assumed to have access to an *oracle* (realized in experiments using a heuristic – see the next section) that can solve weighted classification problems. Given access to such an oracle, oracle efficient algorithms must run in polynomial time. We show that our fairness constrained learning problem is computationally no harder than unconstrained learning by giving such an oracle efficient algorithm (or reduction), and show moreover that its guarantees generalize from in-sample to out-of-sample in the usual way – with respect to both accuracy and the frequency and magnitude of fairness violations. Our algorithm is simple and amenable to implementation, and we use it in our experimental results.

Our Experimental Results

We implement our algorithm and run a set of experiments on the COMPAS recidivism prediction dataset, using fairness constraints elicited from 43 human subjects. We establish that our algorithm converges quickly (even when implemented with fast learning heuristics, rather than "oracles"). We also explore the Pareto curves trading off error and fairness violations for different human subjects, and find empirically that there is a great deal of variability across subjects in terms of their conception of fairness, and in terms of the degree to which their expressed preferences are in conflict with accurate prediction. We find that most of the difficulty in balancing accuracy with the elicited fairness constraints can be attributed to a small fraction of the constraints.

1.2 Related Work

Individual Fairness and Elicitation

Our work is related to existing notions of *individual fairness* like [17, 29] that conceptualize fairness as a set of constraints binding on pairs of individuals. In particular, the notion of metric fairness proposed in [17] is closely related, but distinct from the fairness notions we elicit in this work. In particular: 1) We allow for constraints that require that individual A be treated better than or equal to individual B, whereas metric fairness constraints are symmetric, and only allow constraints of the form that A and B be treated similarly. In this sense our notion is more general; 2) We elicit binary judgments between pairs of individuals, whereas metric fairness is defined as a Lipschitz constraint on a real valued metric. In this sense our notion is more restrictive. Though, we – along with a line of work on classification with pairwise constraints – see merit in pairwise constraints because they "can be relatively easy to collect from human feedback" [49, p. 114].

The most technically related piece of work is Rothblum and Yona [53], who first frame individual fairness in a PAC learning setting and prove similar generalization guarantees to ours for a relaxation of metric fairness. Our conceptual focus is quite different, however: for general learning problems, they prove worst-case hardness results, whereas we derive algorithms in the oracle-efficient model and evaluate them on real elicited user data. The concurrent work of [41] makes a similar observation about guaranteeing fairness with respect to an unknown metric, although their aim is the orthogonal goal of fair representation learning.

Dwork et al. [17] first proposed the notion of individual metric-fairness that we take inspiration from, imagining fairness as a Lipschitz constraint on a randomized algorithm, with respect to some "task-specific metric". Since the original proposal, the question of where the metric should come from has been one of the primary obstacles to its adoption, and the focus of subsequent work. Zemel et al. [67] attempt to automatically learn a representation for the data (and hence, implicitly, a similarity metric) that causes a classifier to label an equal proportion of two protected groups as positive. Kim et al. [35] consider a group-fairness like relaxation of individual metric-fairness, asking that on average, individuals in pre-specified groups are classified with probabilities proportional to the average distance between individuals in those groups. They show how to learn such classifiers given access to an oracle which can evaluate the distance between two individuals according to the metric. Compared to our work, they assume the existence of a fairness metric which can be accessed using a quantitative oracle, and they use this metric to define a statistical rather than individual notion of fairness. Gillen et al. [23] assume access to an oracle which simply identifies fairness violations across pairs of individuals. Under the assumption that the oracle is exactly consistent with a metric in a simple linear class, they give a polynomial time algorithm to compete with the best fair policy in an online linear contextual bandits problem. In contrast to [23], we make essentially no assumptions at all on the structure of the "fairness" constraints. Bechavod et al. [8] generalize the setting of Gillen et al. [23] by making no assumption on the underlying metric and reduce the number of calls to the fairness oracle. Ilvento [28] studies the problem of metric learning with the goal of using only a small number of numeric valued queries, which are hard for human beings to answer, relying more on comparison queries. In contrast with [28], we do not attempt to learn a metric, and instead directly learn a classifier consistent with the elicited pairwise fairness constraints.

Classification with Pairwise Constraints

Stretching back to at least Kleinberg and Tardos [37], there is a line of work that considers classification problems with pairwise constraints which define similarity or dissimilarity between the labels of two data points [49, 7, 49, 68, 6, 57]. Kleinberg and Tardos [37], for example, introduce a classification problem with pairwise equality constraints and a distance metric between pairs. The constraints we elicit differ in that they are asymmetric inequality relations between pairs, rather than equality or metric constraints. Much of this work is conceptually different from ours, as it is concerned with clustering and semi-supervised learning, e.g. [7] or [6].

Preference Elicitation, Social Choice Theory, and Virtual Democracy

Preference elicitation is a well-established area in machine learning [52, 16]. Social choice or voting theory aims to elicit and aggregate people's preferences in order to find a winner or ranking over alternatives (see [50] for an overview). The aim of this work is often to suggest aggregation methods which meet some desirable criteria, such as strategyproofness. *Virtual democracy* is a framework for eliciting normative preferences, constructing a model from these preferences – typically via voting methods – in order to automate moral decision making based on these norms [1, 51, 13, 46, 31, 5, 21]. Our work is conceptually similar: we elicit and learn from moral preferences. However, our work differs in two regards: 1) We focus specifically on fair classification, whereas virtual democracy is concerned with more general moral decisions, e.g. the kind of trolley problem scenarios in the *Moral Machine* experiment [5]; 2) As such, we can incorporate elicited preferences into a fair learning problem rather than using voting methods to aggregate them.

Human Perspectives on Algorithmic Fairness

A number of recent qualitative and HCI works have focused on understanding perspectives on algorithms and fairness from the public [25, 24, 56, 61, 62, 55] and from practitioners [27, 47, 59]. These works suggest that 1) algorithms should incorporate stakeholder input into the fairness principles they use and 2) these fairness principles are context-specific [11, 15, 43, 44, 9, 54]. These works also offer a perspective on how explaining or asking people about fairness influences how they respond [45, 64, 54, 9]. These works provide largely qualitative findings, which may be difficult to translate into specific design implications. Our work complements these prior works by offering a way to easily implement a fair algorithm based on human perspectives.

2 Problem Formulation

Let S denote a set of labeled examples $\{z_i = (x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$ is a feature vector and $y_i \in \mathcal{Y}$ is a label. We will also write $S_X = \{x_i\}_{i=1}^n$ and $S_Y = \{y_i\}_{i=1}^n$. Throughout the paper, we will restrict attention to binary labels, so let $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{P}$ denote the unknown distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ denote a hypothesis class containing binary classifiers $h : \mathcal{X} \to \mathcal{Y}$. We assume that $\mathcal{H}$ contains a constant classifier (which will imply that the "fairness constrained" ERM problem that we define is always feasible). We'll denote classification error of hypothesis h by $err(h, \mathcal{P}) := \Pr_{(x,y)\sim\mathcal{P}}(h(x) \neq y)$ and its empirical classification error by $err(h, S) := \frac{1}{n} \sum_{i=1}^n \mathbb{1}(h(x_i) \neq y_i)$.

We assume there is a set of one or more stakeholders $\mathcal{U}$, such that each stakeholder $u \in \mathcal{U}$ is identified with a set of ordered pairs (x, x') of individuals $C_u \subseteq \mathcal{X}^2$: for each $(x, x') \in C_u$, stakeholder u thinks that x' should be treated as well as x or better, i.e. ideally that for the learned classifier h, the classification $h(x') \geq h(x)$ (we will ask that this hold in expectation if the classifier is randomized, and will relax it in various ways). For each ordered pair (x, x'), let $w_{x,x'}$ be the fraction of stakeholders who would like individual x to be treated as well as x': that is, $w_{x,x'} = \frac{|\{u | (x,x') \in C_u\}|}{|\mathcal{U}|}$. Note that if $(x, x') \in C_u$ and $(x', x) \in C_u$, then the stakeholder wants x and x' to be treated similarly in that ideally $h(x) = h(x')$.

In practice, we will not have direct access to the sets of ordered pairs C_u corresponding to the stakeholders u, but we may ask them whether particular ordered pairs are in this set (see Section 5 for details about how we actually query human subjects). We model this by imagining that we present each stakeholder with a random set of pairs $A \subseteq [n]^2$, and for each ordered pair (x_i, x_j), ask if x_j should not be treated worse than x_i; we learn the set of ordered pairs in $A \cap C_u$ for each u. Define the empirical constraint set $\hat{C}_u = \{(x_i, x_j) \in C_u\}_{\forall (i,j) \in A}$ and $\hat{w}_{x_i x_j} = \frac{|\{u | (x,x') \in \hat{C}_u\}|}{|\mathcal{U}|}$, if $(i, j) \in A$ and 0 otherwise. We write that $\hat{C} = \cup_u \hat{C}_u$. For brevity, we will sometimes write w_{ij} instead of w_{x_i,x_j}. Note that $\hat{w}_{ij} = w_{ij}$ for every $(i, j) \in A$.

Our goal will be to find the distribution over classifiers from $\mathcal{H}$ that minimizes classification error, while satisfying the stakeholders' fairness preferences, captured by the constraints C. To do so, we'll try to find D, a probability distribution over $\mathcal{H}$, that minimizes the training error and satisfies the stakeholders' empirical fairness constraints, $\hat{C}$. For convenience, we denote the expected classification error of D as $err(D, \mathcal{P}) := \mathbb{E}_{h\sim D}[err(h, \mathcal{P})]$ and likewise its expected empirical classification error as $err(D, S) := \mathbb{E}_{h\sim D}[err(h, S)]$. We say that any distribution D over classifiers satisfies (γ, η)-approximate subjective fairness if it is a feasible solution to the following constrained empirical risk minimization problem:

$$\min_{D \in \Delta\mathcal{H}, \alpha_{ij} \geq 0} err(D, S) \tag{1}$$

$$\text{such that } \forall (i, j) \in [n]^2 : \quad \mathop{\mathbb{E}}_{h\sim D}[h(x_i) - h(x_j)] \leq \alpha_{ij} + \gamma \tag{2}$$

$$\sum_{(i,j)\in[n]^2} \frac{\hat{w}_{ij}\alpha_{ij}}{|A|} \leq \eta. \tag{3}$$

This "Fair ERM" problem, whose feasible region we denote by $\Omega(S, \hat{w}, \gamma, \eta)$, has decision variables D and $\{\alpha_{ij}\}$, representing the distribution over classifiers and the "fairness violation" terms for each pair of training points, respectively. The parameters γ and η are constants which represent the two different "knobs" we have at our disposal to quantitatively relax the fairness constraint, in an ℓ_∞ and ℓ_1 sense, respectively.

The parameter γ defines, for any ordered pair (x_i, x_j), the maximum difference between the probabilities that x_i and x_j receive positive labels without constituting a fairness violation. The parameter α_{ij} captures the "excess fairness violation" beyond γ for (x_i, x_j).

The parameter η upper bounds the sum of these allotted excess fairness violation terms α_{ij}, each weighted by the proportion of judges who perceive they ought to be treated similarly $\hat{w}_{ij}$ and normalized with the total number of pairs presented $|A|$. Thus, η bounds the expected degree of dissatisfaction of the panel of stakeholders $\mathcal{U}$, over the random choice of an ordered pair $(x_i, x_j) \in A$ and the randomness of their classification. We iterate over all $(i, j) \in [n]^2$ (not just those in $\hat{C}$) because $\hat{w}_{ij} = 0$ if no judge prefers x_i should be classified as well as x_j.

To better understand γ and η, we consider them in isolation. First, suppose we set $\gamma = 0$. Then, *any* difference in probabilities of positive classification between pairs is deemed a fairness violation. So, if we choose $(D, \{\alpha_{ij}\})$ such that the sum of weighted differences in positive classification probabilities exceeds η, i.e.

$$\sum_{(i,j)\in[n]^2} \frac{\hat{w}_{ij}\, \mathbb{E}_{h\sim D}[h(x_i) - h(x_j)]}{|A|} > \eta,$$

then this is an infeasible solution. Second, suppose that $\eta = 0$. Then, for any $(x_i, x_j) \in C$ (for which $\hat{w}_{ij} > 0$), if the expected difference in labels exceeds γ, i.e. $\mathbb{E}_{h\sim D}[h(x_i) - h(x_j)] > \gamma$, then this is an infeasible solution.

2.1 Fairness Loss

Our goal is to develop an algorithm that will minimize its empirical error $err(D, S)$, while satisfying the empirical fairness constraints $\hat{C}$. The standard VC dimension argument states that empirical classification error will concentrate around the true classification error: we hope to show the same kind of generalization for fairness as well. To do so, we first define fairness loss with respect to our elicited fairness preferences here.

For some fixed randomized hypothesis $D \in \Delta\mathcal{H}$ and w, define γ-fairness loss between an ordered pair as $\Pi_{D,w,\gamma}((x, x')) = w_{x,x'} \max(0, \mathbb{E}_{h\sim D}[h(x) - h(x')] - \gamma)$. For a set of pairs $M \subset \mathcal{X} \times \mathcal{X}$, the γ-fairness loss of M is defined to be: $\Pi_{D,w,\gamma}(M) = \frac{1}{|M|} \sum_{(x,x')\in M} \Pi_{D,w,\gamma}((x, x'))$. This is the expected degree to which the difference in classification probability for a randomly selected pair exceeds the allowable budget γ, weighted by the fraction of stakeholders who think that x' should be treated as well as x. By construction, the empirical fairness loss is bounded by η (i.e. $\Pi_{D,w,\gamma}(M) \le \sum_{ij} \frac{\hat{w}_{ij}\alpha_{ij}}{|A|} \le \eta$), and we show in Section 4, the empirical fairness should concentrate around the true fairness loss $\Pi_{D,w,\gamma}(\mathcal{P}) := \mathbb{E}_{x,x'\sim\mathcal{P}^2}[\Pi_{D,w,\gamma}(x, x')]$.

2.2 Cost-sensitive Classification

In our algorithm, we will make use of a cost-sensitive classification (CSC) oracle. An instance of CSC problem can be described by a set of costs $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ and a hypothesis class, $\mathcal{H}$. Costs c_i^0 and c_i^1 correspond to the cost of labeling x_i as 0 and 1 respectively. Invoking a CSC oracle on $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ returns a hypothesis h^* such that $h^* \in \operatorname{argmin}_{h\in\mathcal{H}} \sum_{i=1}^n \left(h(x_i)c_i^1 + (1 - h(x_i))\, c_i^0\right)$. We say that an algorithm is *oracle-efficient* if it runs in polynomial time assuming access to a CSC oracle.

3 Empirical Risk Minimization

In this section, we give an oracle-efficient algorithm whose pseudocode is shown in Algorithm 1 for approximately solving our (in-sample) constrained empirical risk minimization problem. Details are deferred to the full version of this paper which is available on arXiv [30]. We prove the following theorem:

Algorithm 1 No-Regret Dynamics.

```
Input: training examples $\{x_i, y_i\}_{i=1}^n$, bounds $C_\lambda$ and $C_\tau$, time horizon $T$, step sizes $\mu_\lambda$ and $\{\mu_\tau^t\}_T^{t=1}$,
Set $\theta_1^0 = \mathbf{0} \in \mathbb{R}^{n^2}$
Set $\tau^0 = 0$
for $t = 1, 2, \dots, T$ do
    Set $\lambda_{ij}^t = C_\lambda \frac{\exp \theta_{ij}^{t-1}}{1+\sum_{i',j' \in [n]^2} \exp \theta_{i'j'}^{t-1}}$ for all pairs $(i,j) \in [n]^2$
    Set $\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu_\tau^t \left(\frac{1}{|A|}\sum_{i,j} w_{ij}\alpha_{ij}^{t-1} - \eta\right)\right)$
    $D^t, \alpha^t \leftarrow \mathrm{BEST}_\rho(\lambda^t, \tau^t)$
    for $(i,j) \in [n]^2$ do
        $\theta_{ij}^t = \theta_{ij}^{t-1} + \mu_\lambda^{t-1}\left(\mathbb{E}_{h\sim D^t}[h(x_i) - h(x_j)] - \alpha_{ij}^t - \gamma\right)$

Output: $\frac{1}{T}\sum_{t=1}^T D^t$, $\frac{1}{T}\sum_{t=1}^T \alpha^t$
```

▶ **Theorem 1.** *Fix parameters ν, C_τ, C_λ that serve to trade off running time with approximation error. There is an efficient algorithm that makes $T = \left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{\nu}\right)^2$ CSC oracle calls and outputs a solution $(\hat{D}, \hat{\alpha})$ with the following guarantee. The objective value is approximately optimal:*

$$err(\hat{D}, S) \leq \min_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D,S) + 2\nu.$$

And the constraints are approximately satisfied: $\mathbb{E}_{h\sim\hat{D}}[h(x_i)-h(x_j)] \leq \hat{\alpha}_{ij}+\gamma+\frac{1+2\nu}{C_\lambda}, \forall(i,j) \in [n]^2$ and $\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\hat{\alpha}_{ij} \leq \eta + \frac{1+2\nu}{C_\tau}$.

3.1 Outline of the Solution

We frame the problem of solving our constrained ERM problem (equations (1) through (3)) as finding an approximate equilibrium of a zero-sum game between a primal player and a dual player, trying to minimize and maximize respectively the Lagrangian of the constrained optimization problem.

The Lagrangian for our optimization problem is

$$\begin{aligned}\mathcal{L}(D,\alpha,\lambda,\tau) = err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(\mathop{\mathbb{E}}_{h\sim D}[h(x_i)-h(x_j)] - \alpha_{ij} - \gamma\right) \\ + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right)\end{aligned}$$

For the constraint in equation (2), corresponding to the γ-fairness violation for each ordered pair of individuals (x_i, x_j), we introduce a dual variable λ_{ij}. For the constraint (3), which corresponds to the η-fairness violation over all pairs of individuals, we introduce a dual variable of τ. For brevity, we define vectors $\lambda \in \Lambda$ and α which are made up of all the multipliers λ_{ij} and the excess fairness violation allotments α_{ij}, respectively. The primary player's action space is $(D,\alpha) \in (\Delta\mathcal{H}, [0,1]^{n^2})$, and the dual player's action space is $(\lambda,\tau) \in (\mathbb{R}^{n^2}, \mathbb{R})$.

Solving our constrained ERM problem is equivalent to finding a minmax equilibrium of $\mathcal{L}$:

$$\operatorname*{argmin}_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} \mathit{err}(D,S) = \operatorname*{argmin}_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \max_{\lambda\in\mathbb{R}^{n^2},\tau\in\mathbb{R}} \mathcal{L}(D,\alpha,\lambda,\tau)$$

Because $\mathcal{L}$ is linear in terms of its parameters, Sion's minimax theorem [58] gives us

$$\min_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \max_{\lambda\in\mathbb{R}^{n^2},\tau\in\mathbb{R}} \mathcal{L}(D,\alpha,\lambda,\tau) = \max_{\lambda\in\mathbb{R}^{n^2},\tau\in\mathbb{R}} \min_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \mathcal{L}(D,\alpha,\lambda,\tau).$$

By a classic result of Freund and Schapire [22], one can compute an approximate equilibrium by simulating "no-regret" dynamics between the primal and dual player. "No-regret" meaning that the average *regret* –or difference between our algorithm's plays and the single best play in hindsight– is bounded above by a term that converges to zero with increasing rounds.

In our case, we define a zero-sum game wherein the primary player's plays from action space $(D,\alpha)\in(\Delta\mathcal{H},[0,1]^{n^2})$, and the dual player's plays from action space $(\lambda,\tau)\in(\mathbb{R}^{n^2}_{\geq 0},\mathbb{R}_{\geq 0})$. In any given round t, the dual player plays first and the primal second. The primal player can simply best respond to the dual player (see Algorithm 2).

However, since the dual player plays first, they cannot simply best respond to the primal player's action. The dual player has to anticipate the primal player's best response in order to figure out what to play. Ideally, the dual player would enumerate every possible primal play and calculate the best dual response. However, this is intractable. So, the dual player updates dual variables $\{\lambda,\tau\}$ according to *no-regret* learning algorithms (exponentiated gradient descent [36] and online gradient descent [69], respectively).

The time-averaged play of both players converges to an approximate equilibrium of the zero-sum game, where the approximation is controlled by the regret of the dual player. This approximate equilibrium corresponds to an approximate saddle point for the Lagrangian $\mathcal{L}$, which is equivalent to an approximate solution to the Fair ERM problem.

We organize the rest of this section as follows. First, for simplicity, we show how the primal player updates $\{D,\alpha\}$ (even though the dual player plays first). Second, we show how the dual player updates $\{\lambda,\tau\}$. Finally, we prove that these updates are no-regret and relate the regret of the dual player to the approximation of the solution to the Fair ERM problem.

3.2 The Primal Player's Best Response

In each round t, given the actions chosen by the dual player (λ^t,τ^t), the primal player needs to best respond by choosing (D^t,α^t) such that $(D^t,\alpha^t)\in\operatorname{argmin}_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}}\mathcal{L}(D,\alpha,\lambda^t,\tau^t)$. We can separate the optimization problem into two as shown above in Algorithm 2: one optimization over hypothesis D and one over violation factor α. As for D^t, the primal player can update the hypothesis D by leveraging a CSC oracle. Given λ^t, we can set the costs as follows:

$$c_i^0 = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(y_i\neq 0)\right], \qquad c_i^1 = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}\left(y_i\neq 1\right)\right] + (\lambda_{ij}^t - \lambda_{ji}^t).$$

Then, $D^t = h^t = CSC\left(\{(x_i,c_i^0,c_i^1)\}_{i=1}^n\right)$ (we note that the best response is always a deterministic classifier h^t).

As for α^t, we can show that the primal player sets $\alpha_{ij}^t = 1$ if $\tau^t\frac{w_{ij}}{|A|} - \lambda_{ij}^t \leq 0$ and 0 otherwise. We defer its derivation and proofs to the full version of this paper which is available on arXiv [30].

Algorithm 2 Best Response, $BEST_\rho(\lambda, \tau)$, for the primal player.

```
Input: training examples $S = \{x_i, y_i\}_{i=1}^n$, $\lambda \in \Lambda$, $\tau \in \mathcal{T}$, CSC oracle $CSC$
for $i = 1, \dots, n$ do
    if $y_i = 0$ then
        Set $c_i^0 = 0$
        Set $c_i^1 = \frac{1}{n} + \sum_{j \neq i} \lambda_{ij} - \lambda_{ji}$
    else
        Set $c_i^0 = \frac{1}{n}$
        Set $c_i^1 = \sum_{j \neq i} \lambda_{ij} - \lambda_{ji}$
$D = CSC(S, c)$
for $(i, j) \in [n]^2$ do
    $\alpha_{ij} = \begin{cases} 1 : & \tau \frac{w_{ij}}{|A|} - \lambda_{ij} \leq 0 \\ 0 : & \tau \frac{w_{ij}}{|A|} - \lambda_{ij} > 0. \end{cases}$
Output: $D, \alpha$
```

3.3 The Dual Player's No-regret Updates

In order to reason about convergence, we need to restrict the dual player's action space to lie within a bounded ℓ_1 ball, defined by the parameters C_τ and C_λ that appear in our theorem – and serve to trade off running time with approximation quality: $\Lambda = \left\{\lambda \in \mathbb{R}_+^{n^2} : \|\lambda\|_1 \leq C_\lambda\right\}, \mathcal{T} = \{\tau \in \mathbb{R}_+ : \|\tau\|_1 \leq C_\tau\}$. The dual player will use exponentiated gradient descent [36] to update λ and online gradient descent [69] to update τ, where the reward function will be defined as: $r_\lambda(\lambda^t) = \sum_{(i,j)\in[n]^2} \lambda_{ij}^t \left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right)$ and $r_\lambda(\tau^t) = \tau^t \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right)$. We defer its derivation and proofs to the full version of this paper which is available on arXiv [30].

4 Generalization

In this section, we show that fairness loss generalizes out-of-sample. (Error generalization follows from the standard VC-dimension bound, which – because it is a uniform convergece statement is unaffected by the addition of fairness constraints. See the supplement for the standard statement.)

Proving that the fairness loss generalizes doesn't follow immediately from a standard VC-dimension argument for several reasons: it is not linearly separable, but defined as an average over non-disjoint *pairs* of individuals in the sample. The difference between empirical fairness loss and true fairness loss of a randomized hypothesis $D \in \Delta\mathcal{H}$ is also a non-convex function of the supporting hypotheses h, and so it is not sufficient to prove a uniform convergence bound merely for the base hypotheses in our hypothesis class $\mathcal{H}$. We circumvent these difficulties by making use of an ε-net argument, together with an application of a concentration inequality, and an application of Sauer's lemma. Briefly, we show that with respect to fairness loss, the continuous set of distributions over classifiers have an ε-net of sparse distributions. Using the two-sample trick and Sauer's lemma, we can bound the number of such sparse distributions. The end result is the following generalization theorem:

▶ **Theorem 2.** *Let S consists of n i.i.d points drawn from $\mathcal{P}$ and let M represent a set of m pairs randomly drawn from $S \times S$. Then we have:*

$$\Pr_{\substack{S\sim\mathcal{P}^n\\ M\sim(S\times S)^m}}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(M) - \mathop{\mathbb{E}}_{(x,x')\sim\mathcal{P}^2}\left[\Pi_{D,w,\gamma}(x,x')\right]\right| > 2\varepsilon\right)$$
$$\leq \left(8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\varepsilon^2}{32}\right) + \left(\frac{e\cdot 2n}{d}\right)^{dk'}\exp\left(-8m\varepsilon^2\right)\right),$$

where $k' = \frac{2\ln(2m)}{\varepsilon^2} + 1$, $k = \frac{\ln(2n^2)}{8\varepsilon^2} + 1$, and d is the VC-dimension of $\mathcal{H}$.

To interpret this theorem, note that the right hand side (the probability of a failure of generalization) begins decreasing exponentially fast in the data and fairness constraint sample parameters n and m as soon as $n \geq \Omega(d\log(n)\log(n/d))$ and $m \geq \Omega(d\log(m)\log(n/d))$.

5 A Behavioral Study

The framework and algorithm we have provided can be viewed as a tool to elicit and enforce a notion of fairness defined by a collection of stakeholders. In this section, we describe preliminary results from a human-subject study we performed in which pairwise fairness preferences were elicited and enforced by our algorithm. We note that the subjects included in our empirical study were not stakeholders affected by the algorithm we used (the COMPAS algorithm). Thus, our results should not be interpreted as cogent for any policy modifications to the COMPAS algorithm. We instead report our empirical findings primarily to showcase the performance of our algorithm and to act as a template for what should be reported if our framework were applied with relevant stakeholders (for example, if fairness preferences about COMPAS data were elicited from inmates).[4]

5.1 Data

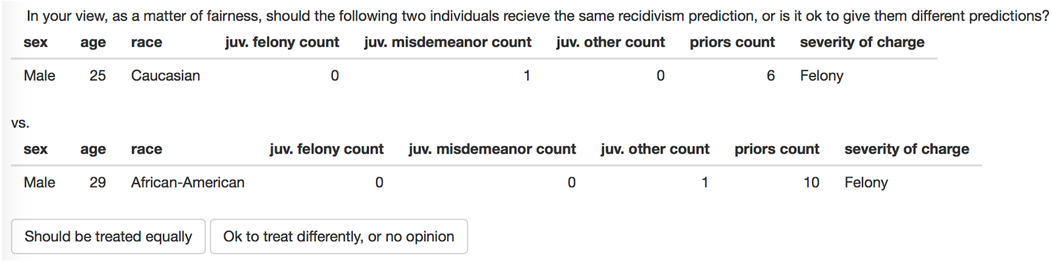
Our study used the COMPAS recidivism data gathered by ProPublica [5] in their celebrated analysis of Northepointe's risk assessment algorithm [42]. This data consists of defendants from Broward County in Florida between 2013 to 2014. For each defendant the data consists of sex (male, female), age (18-96), race (African-American, Caucasian, Hispanic, Asian, Native American), juvenile felony count, juvenile misdemeanor count, number of other juvenile offenses, number of prior adult criminal offenses, the severity of the crime for which they were incarcerated (felony or misdemeanor), as well as the outcome of whether or not they did in fact recidivate. Recidivism is defined as a new arrest within 2 years, not counting traffic violations and municipal ordinance violations.

5.2 Subjective Fairness Elicitation

We implemented our fairness framework via a web app that elicited subjective fairness notions from 43 undergraduates at a major research university. After reading a document describing the data and recidivism prediction task, each subject was presented with 50 randomly chosen

[4] We omit such an empirical study due to the difficulty of accessing such stakeholders and leave this for future work.

[5] The data can be accessed on ProPublica's Github page here. We cleaned the data as in the ProPublica study, removing any records with missing data. This left 5829 records, where the base rate of two-year recidivism was 46%.



In your view, as a matter of fairness, should the following two individuals recieve the same recidivism prediction, or is it ok to give them different predictions?

sex	age	race	juv. felony count	juv. misdemeanor count	juv. other count	priors count	severity of charge
Male	25	Caucasian	0	1	0	6	Felony

vs.

sex	age	race	juv. felony count	juv. misdemeanor count	juv. other count	priors count	severity of charge
Male	29	African-American	0	0	1	10	Felony

Should be treated equally
Ok to treat differently, or no opinion

Figure 1 Screenshot of sample subjective fairness elicitation question posed to human subjects.

pairs of records from the COMPAS data set and asked whether in their opinion the two individuals should treated (predicted) equally or not. Importantly, the subjects were shown only the features for the individuals, and not their actual recidivism outcomes, since we sought to elicit subjects' fairness notions regarding the predictions of those outcomes. While absolutely no guidance was given to subjects regarding fairness, the elicitation framework allows for rich possibilities. For example, subjects could choose to ignore demographic factors or criminal histories entirely if they liked, or a subject who believes that minorities are more vulnerable to overpolicing could discount their criminal histories relative to Caucasians in their pairwise elicitations.

For each subject, the pairs they identified to be treated equally were taken as constraints on error minimization with respect to the actual recidivism outcomes over the entire COMPAS dataset, and our algorithm was applied to solve this constrained optimization problem, using a linear threshold heuristic as the underlying learning oracle [33]. We ran our algorithm with $\eta = 0$ and variable γ in Equations (1) through (3), which represents the strongest enforcement of subjective fairness – the difference in predicted values must be at most γ on *every* pair selected by a subject. Because the issues we are most interested in here (convergence, tradeoffs with accuracy, and heterogeneity of fairness preferences) are orthogonal to generalization – and because we prove VC-dimension based generalization theorems – for simplicity, the results we report are in-sample.

5.3 Results

Since our algorithm relies on a learning heuristic for which worst-case guarantees are not possible, the first empirical question is whether the algorithm converges rapidly on the behavioral data. We found that it did so consistently; a typical example is Figure 2a, where we show the trajectories of model error vs. fairness violation for a particular subject's data for variable values of the input γ (horizontal lines). After 1000 iterations, the algorithm has converged to the optimal errors subject to the allowed γ.

Perhaps the most basic behavioral questions we might ask involve the extent and nature of subject variability. For example, do some subjects identify constraint pairs that are much harder to satisfy than other subjects? And if so, what factors seem to account for such variation?

Figure 2b shows that there is indeed considerable variation in subject difficulty. For each of the 43 subjects, we have plotted the error vs. fairness violation Pareto curves obtained by varying γ from 0 (pairs selected by subjects must have identical probabilistic predictions of recidivism) to 1.0 (no fairness enforced whatsoever). Since our model space is closed under probabilistic mixtures, the worst-case Pareto curve is linear, obtained by all mixtures of the error-optimal model and random predictions. Easier constaint sets are more convex. We see

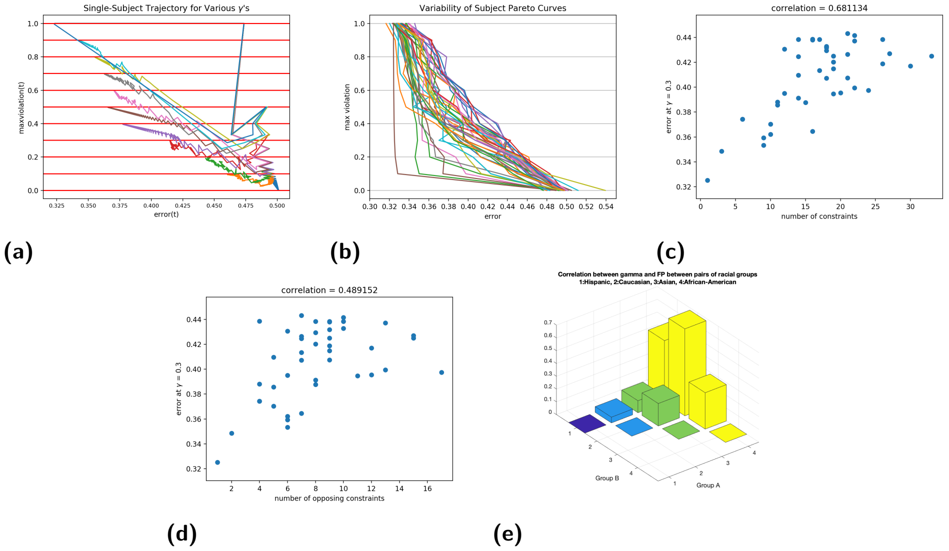


Figure 2 (a) Sample algorithm trajectory for a particular subject at various γ. (b) Sample subjective fairness Pareto curves for a sample of subjects. (c) Scatterplot of number of constraints specified and number of opposing constraints vs. error at $\gamma = 0.3$. (d) Scatterplot of number of constraints where the true labels are different vs. error at $\gamma = 0.3$. (e) Correlation between false positive rate difference and γ for racial groups.

in the figure that both extremes are exhibited behaviorally – some subjects yield linear or near-linear curves, while others permit huge reductions in unfairness for only slight increases in error, and virtually all the possibilities in between are realized as well. [6]

Since each subject was presented with 50 random pairs and was free to constrain as many or as few as they wished, it is natural to wonder if the variation in difficulty is explained simply by the number of constraints chosen. In Figure 2c we show a scatterplot of the the number of constraints selected by a subject (x axis) versus the error obtained (y axis) for $\gamma = 0.3$ (an intermediate value that exhibits considerable variation in subject error rates) for all 43 subjects. While we see there is indeed strong correlation (approximately 0.69), it is far from the case that the number of constraints explains all the variability. For example, amongst subjects who selected approximately 16 constraints, the resulting error varies over a range of nearly 8%, which is over 40% of the range from the optimal error (0.32) to the worst fairness-constrained error (0.5). More surprisingly, when we consider only the "opposing" constraints, pairs of points with different true labels, the correlation (0.489) seems to be weaker. Enforcing a classifier to predict similarly on a pair of points with different true labels should increase the error, and yet, it is less correlated with error than the raw number of constraints. This suggests that the variability in subject difficulty is due to the nature of the constraints themselves rather than their number or disagreement with the true labels.

It is also interesting to consider the collective force of the 1432 constraints selected by all 43 subjects together, which we can view as a "fairness panel" of sorts. Given that there are already individual subjects whose constraints yield the worst-case Pareto curve, it is unsurprising that the collective constraints do as well. But we can exploit the flexibility of our optimization framework in Equations (1) through constraint (3), and let $\gamma = 0.0$ and vary only η, thus giving the learner discretion in which subjects' constraints to discount or discard at a given budget η. In doing so we find that the unconstrained optimal error

[6] The slight deviations from true convexity are due to approximate rather than exact convergence.

can be obtained while having the average (exact) pairwise constraint be violated by only roughly 25%, meaning roughly that only 25% of the collective constraints account for all the difficulty.

Finally, we can investigate the extent to which behavioral subjective fairness notions align with more standard statistical fairness definitions, such as equality of false positive rates. For instance, for each subject and a pair of racial groups, we take the absolute difference in false positive rates of the classifier at $\gamma \in \{0.0, 0.1, \ldots, 1.0\}$ and calculate the correlation coefficient between realized values of γ (which measure violation of subjective unfairness) and the false positive rate differences. Figure 2e shows the average correlation coefficient across subjects for each pair of racial groups. We note that subjective fairness correlates with a smaller gap between the false positive rates across Caucasians and African Americans: but correlates substantially less for other pairs of racial groups.

We leave a more complete investigation of our behavioral study for future work, including the detailed nature of subject variability and further comparison of behavioral subjective fairness to standard algorithmic fairness notions.

6 Discussion and Limitations

We provide a framework to involve non-technical stakeholders into the process of defining algorithmic fairness. Our approach offers a means for stakeholders to encode their more nuanced, contextual principles of fairness into a model. By eliciting pairwise fairness preferences, our approach is designed to be easily understood by laypeople, even if they do not understand the technicalities of the learning algorithm. We provide theoretical guarantees, as well as preliminary experiments to demonstrate the functionality of our algorithm.

Here, we anticipate a criticism: biases may be embedded into the model if the stakeholders we elicit from are biased. To address this, we clarify that our approach aims to easily elicit and operationalize what stakeholders think is fair and unbiased. As such, if stakeholders truthfully report their preferences, then our approach should produce a model which these stakeholders believe is unbiased. In this sense, we consider what is biased and what is fair in our context to be subjective. While this point of view may be uncomfortable within the algorithmic fairness community, it is a well-established argument within ethics and normative economics [38]. Furthermore, if the design goal is for the model to achieve some objective standard of neutrality, then we believe this should be addressed at the level of choosing which stakeholders to elicit from and how they are elicited – see Section 1.2 for related qualitative and HCI works which consider how to elicit fairness preferences from people. We consider this to be an important direction for future work.[7]

References

1 Matthew Adler. Aggregating moral preferences. *Economics and Philosophy*, 32:283–321, 2016.

2 Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69, 2018.

3 Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 60–69, 2018. URL: http://proceedings.mlr.press/v80/agarwal18a.html.

[7] See, for example, a follow-up interview study which elicits pairwise fairness beliefs about risk assessment tools used in the Allegheny County child welfare system [12].

4 Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 120–129, 2019. URL: `http://proceedings.mlr.press/v97/agarwal19d.html`.

5 Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. The moral machine experiment. *Nature*, 563:59–64, 2018.

6 Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *ICML*, 2018.

7 Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining, (SDM-2004), pp. , Lake Buena Vista, FL*, pages 333—-344, 2004.

8 Yahav Bechavod, Christopher Jung, and Steven Z. Wu. Metric-free individual fairness in online learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: `https://proceedings.neurips.cc/paper/2020/hash/80b618ebcac7aa97a6dac2ba65cb7e36-Abstract.html`.

9 Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. "It's reducing a human being to a percentage": Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 377. ACM, 2018.

10 Lawrence Blum. *Moral Perception and Particularity.* Cambridge University Press, 1994.

11 Anna Brown, Alexandra Chouldechova, Emily Putnam-Hornstein, Andrew Tobin, and Rhema Vaithianathan. Toward algorithmic accountability in public services: A qualitative study of affected community perspectives on algorithmic decision-making in child welfare services. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 41, 2019. `doi:10.1145/3290605.3300271`.

12 Hao-Fei Cheng, Logan Stapleton, Ruiqi Wang, Paige Bullock, Alexandra Chouldechova, Zhiwei Steven Wu, and Haiyi Zhu. Soliciting stakeholders' fairness notions in child maltreatment predictive systems. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2021. `arXiv:2102.01196`.

13 Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, and Max Kramer. Moral decision making frameworks for artificial intelligence. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2018.

14 Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *CoRR*, abs/1808.00023, 2018. `arXiv:1808.00023`.

15 Jonathan Dodge, Q. Vera Liao, Yunfeng Zhang, Rachel K. E. Bellamy, and Casey Dugan. Explaining models: An empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 275–285, New York, NY, USA, 2019. ACM. `doi:10.1145/3301275.3302310`.

16 Carmel Domshlak, Eyke Hüllermeier, Souhila Kaci, and Henri Prade. Preferences in ai: An overview. *Artificial Intelligence*, 175(7):1037—-1052, 2011. `doi:10.1016/j.artint.2011.03.004`.

17 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*, pages 214–226. ACM, 2012.

18 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226, 2012. `doi:10.1145/2090236.2090255`.

19 Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

20 Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

21 Rachel Freedman, Jana Schaich Borg, Walter Sinnott-Armstrong, John P. Dickerson, and Vincent Conitzer. Adapting a kidney exchange algorithm to align with human values. *Artificial Intelligence*, 283:103261, 2020.

22 Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996.

23 Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems*, pages 2600–2609, 2018.

24 Nina Grgic-Hlaca, Elissa M. Redmiles, Krishna P. Gummadi, and Adrian Weller. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 903–912. International World Wide Web Conferences Steering Committee, 2018. doi:10.1145/3178876.3186138.

25 Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

26 Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323, 2016. URL: http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning.

27 Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miroslav Dudík, and Hanna M. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 600, 2019. doi:10.1145/3290605.3300830.

28 C Ilvento. Metric learning for individual fairness. Manuscript submitted for publication, 2019.

29 Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

30 Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. An algorithmic framework for fairness elicitation. *arXiv preprint*, 2019. arXiv:1905.10660.

31 Anson Kahng, Min Kyung Lee, Ritesh Noothigattu, Ariel D. Procaccia, and Christos-Alexandros Psomas. Statistical foundations of virtual democracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3173–3182, 2019.

32 Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

33 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2569–2577, 2018.

34 Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 2569–2577, 2018. URL: http://proceedings.mlr.press/v80/kearns18a.html.

35 Michael Kim, Omer Reingold, and Guy Rothblum. Fairness through computationally-bounded awareness. In *Advances in Neural Information Processing Systems*, pages 4842–4852, 2018.

36 Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63, 1997.

37 Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. In *P40th Annual Symposium on Foundations of Computer Science, New York, NY, USA*, pages 14—-23, 1999. `doi:10.1109/SFFCS.1999.814572`.

38 James Konow. Is fairness in the eye of the beholder? an impartial spectator analysis of justice. *Social Choice and Welfare*, 33:101—-127, 2009.

39 Felicitas Kraemer, Kees van Overveld, and Martin Peterson. Is there an ethics of algorithms? *Ethics and Information Technology*, 13:251—-260, 2011.

40 Bogdan Kulynych, David Madras, Smitha Milli, Inioluwa Deborah Raji, Angela Zhou, and Richard Zemel. Participatory approaches to machine learning. International Conference on Machine Learning Workshop, 2020.

41 Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. *CoRR*, abs/1907.01439, 2019. `arXiv:1907.01439`.

42 Jeff Larson, Julia Angwin, Lauren Kirchner, and Surya Mattu. How we analyzed the compas recidivism algorithm, March 2019. URL: `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

43 Min Kyung Lee. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1):2053951718756684, 2018.

44 Min Kyung Lee and Su Baykal. Algorithmic mediation in group decisions: Fairness perceptions of algorithmically mediated vs. discussion-based social division. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1035–1048. ACM, 2017.

45 Min Kyung Lee, Ji Tae Kim, and Leah Lizarondo. A human-centered approach to algorithmic services: Considerations for fair and motivating smart community service management that allocates donations to non-profit organizations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3365–3376. ACM, 2017.

46 Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. Webuildai: Participatory framework for algorithmic governance. *Proc. ACM Hum. Comput. Interact.*, 3(CSCW):181:1–181:35, 2019.

47 John Monahan, Anne Metz, and Brandon L Garrett. Judicial appraisals of risk assessment in sentencing. *Virginia Public Law and Legal Theory Research Paper*, No. 2018-27, 2018.

48 Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, 2018.

49 Nam Nguyen and Rich Caruana. Improving classification with pairwise constraints: A margin-based approach. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 113–124, 2008.

50 Noam Nisan. Introduction to mechanism design (for computer scientists). In *N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, Algorithmic Game Theory*. Cambridge University Press, 2007.

51 Ritesh Noothigattu, Snehalkumar (Neil) S. Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel D. Procaccia. A voting-based system for ethical decision making. In *Proceedings of the 32nd Conference on Artificial Intelligence, (AAAI)*, pages 1587–1594, 2018.

52 Gabriella Pigozzi, Alexis Tsoukiàs, and Paolo Viappiani. Preferences in artificial intelligence. *Annals of Mathematics and Artificial Intelligence*, 77:361–401, 2016.

53 Guy N Rothblum and Gal Yona. Probably approximately metric-fair learning. *arXiv preprint*, 2018. `arXiv:1803.03242`.

54 Debjani Saha, Candice Schumann, Duncan C. McElfresh, John P. Dickerson, Michelle L. Mazurek, and Michael Carl Tschantz. Measuring non-expert comprehension of machine learning fairness metrics. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Vienna, Austria, July 12–18, 2020*, 2020.

55 Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C. Parkes, and Yang Liu. How do fairness definitions fare?: Examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 99—-106. ACM, 2019.

56 Nicholas Scurich and John Monahan. Evidence-based sentencing: Public openness and opposition to using gender, age, and race as risk factors for recidivism. *Law and Human Behavior*, 40(1):36, 2016.

57 Takuya Shimada, Han Bao, Issei Sato, and Masashi Sugiyama. Classification from pairwise similarities/dissimilarities and unlabeled data via empirical risk minimization. In *Neural Computation*, pages 1–35, 2021. doi:10.1162/neco_a_01373.

58 Maurice Sion et al. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.

59 Michael Veale, Max Van Kleek, and Reuben Binns. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–14, 2018.

60 Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

61 AJ Wang. Procedural justice and risk-assessment algorithms, 2018. doi:10.2139/ssrn.3170136.

62 Ruotong Wang, F Maxwell Harper, and Haiyi Zhu. Factors influencing perceived fairness in algorithmic decision-making: Algorithm outcomes, development procedures, and individual differences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

63 Pak-Hang Wong. Democratizing algorithmic fairness. *Philosophy & Technology*, 33:225–244, 2020.

64 Allison Woodruff, Sarah E Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. A qualitative exploration of perceptions of algorithmic fairness. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 656. ACM, 2018.

65 Ariana Yaptangco. Male tennis pros confirm serena's penalty was sexist and admit to saying worse on the court. *Elle*, 2017. URL: http://www.elle.com/culture/a23051870/male-tennis-pros-confirm-serenas-penalty-was-sexist-and-admit-to-saying-worse-on-the-court/.

66 Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web, WWW*, pages 1171–1180. ACM, 2017.

67 Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

68 H. Zeng and Y. Cheung. Semi-supervised maximum margin clustering with pairwise constraints. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):926–939, 2012. doi:10.1109/TKDE.2011.68.

69 Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

A Missing Details And Proofs

We defer all the missing details and proofs to the full version of this paper which is available on arXiv [30].

On the Computational Tractability of a Geographic Clustering Problem Arising in Redistricting

Vincent Cohen-Addad
CNRS and Sorbonne Université, Paris, France

Philip N. Klein
Brown University, Providence, RI, USA

Dániel Marx ✉
CISPA Helmholtz Center for Information Security, Saarland Informatics Campus, Germany

Archer Wheeler
Brown University, Providence, RI, USA

Christopher Wolfram
Brown University, Providence, RI, USA

Abstract

Redistricting is the problem of dividing up a state into a given number k of regions (called *districts*) where the voters in each district are to elect a representative. The three primary criteria are: that each district be connected, that the populations of the districts be equal (or nearly equal), and that the districts are "compact". There are multiple competing definitions of compactness, usually minimizing some quantity.

One measure that has been recently been used is number of *cut edges*. In this formulation of redistricting, one is given atomic regions out of which each district must be built (e.g., in the U.S., census blocks). The populations of the atomic regions are given. Consider the graph with one vertex per atomic region and an edge between atomic regions with a shared boundary of positive length. Define the weight of a vertex to be the population of the corresponding region. A districting plan is a partition of vertices into k pieces so that the parts have nearly equal weights and each part is connected. The districts are considered compact to the extent that the plan minimizes the number of edges crossing between different parts.

There are two natural computational problems: find the most compact districting plan, and sample districting plans (possibly under a compactness constraint) uniformly at random.

Both problems are NP-hard so we consider restricting the input graph to have branchwidth at most w. (A planar graph's branchwidth is bounded, for example, by its diameter.) If both k and w are bounded by constants, the problems are solvable in polynomial time. In this paper, we give lower and upper bounds that characterize the complexity of these problems in terms of parameters k and w. For simplicity of notation, assume that each vertex has unit weight. We would ideally like algorithms whose running times are of the form $O(f(k,w)n^c)$ for some constant c independent of k and w (in which case the problems are said to be *fixed-parameter tractable* with respect to those parameters). We show that, under standard complexity-theoretic assumptions, no such algorithms exist. However, the problems *are* fixed-parameter tractable with respect to each of these parameters individually: there exist algorithms with running times of the form $O(f(k)n^{O(w)})$ and $O(f(w)n^{k+1})$. The first result was previously known. The new one, however, is more relevant to the application to redistricting, at least for coarse instances. Indeed, we have implemented a version of the algorithm and have used to successfully find optimally compact solutions to all redistricting instances for France (except Paris, which operates under different rules) under various population-balance constraints. For these instances, the values for w are modest and the values for k are very small.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases redistricting, algorithms, planar graphs, lower bounds

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.3

Related Version A version without discussion of the implementation is at:
Previous Version: `http://https://arxiv.org/abs/2009.00188`

Funding *Philip N. Klein*: Supported by National Science Foundation grant CCF-1841954.
Dániel Marx: Supported by European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH.
Archer Wheeler: Supported by National Science Foundation grant CCF-1841954.

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 3; pp. 3:1–3:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

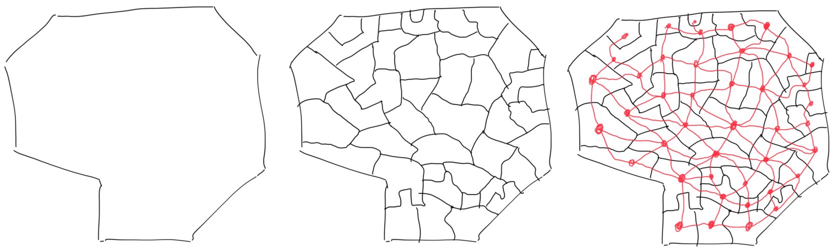


Figure 1 On the left is an imaginary state/department. In the middle, the state is subdivided into smaller regions (*atoms*), e.g. census tracts. On the right, the planar dual is shown. Each atomic region is represented by a node. (There is also a node for the single infinite region outside the state boundary but here we ignore that node here.) For each maximal contiguous boundary segment between a pair of atomic regions, the planar dual has an edge between the corresponding pair of nodes.

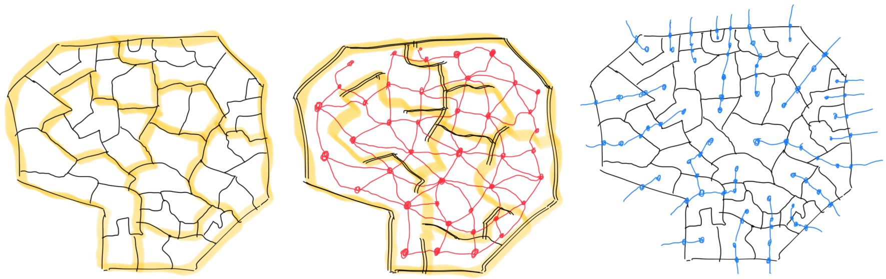


Figure 2 The figure on the left shows an example of a districting plan with seven districts. Each district is the union of several atomic regions. The figure in the middle depicts the districting plan superimposed on the planar dual, showing that it corresponds to a partition of the atoms into connected parts; the cost of the solution is the sum of costs of edges of the dual that cross between different parts. In this paper, a districting plan is compact to the extent that this sum of costs is small. The figure on the right illustrates a breadth-first search in the radial graph of the graph G of atomic regions. As stated in Section 2.2, the *radial graph* of G has a node for every vertex of G and a node for every face of G, and an edge between a vertex-node and a face-node if the vertex lies on the face's boundary. This diagram shows that every face is reachable from the outer face within six hops in the *radial graph* of the graph G of atomic regions. This implies that the branchwidth of G and of its dual are at most six.

1 Introduction

For an undirected planar graph G with vertex-weights and a positive integer k, a *connected partition* of the vertices of G is a partition into parts each of which induces a connected subgraph. If G is equipped with nonnegative integral vertex weights and $[L, U)$ is an interval we say such a partition has *part-weight* in $[L, U)$ if the sum of weights of each part lies in the interval. If G is equipped with nonnegative edge costs, we say the *cost* of such a partition is the sum of costs of edges uv where u and v lie in different parts.

Consider the following computational problems:

- *optimization:* Given a planar graph G with vertex weights and edge costs, a number k, and a weight interval $[L, U)$, find the minimum cost of a partition into k connected parts with part-weight in $[L, U)$.
- *sampling:* Given in addition a number C, generate uniformly at random a cost-C partition into k connected parts with part-weight in $[L, U)$.

These problem arise in political redistricting. Each vertex represents a small geographical region (such as a *census block* or *census tract* or *county*), and its weight represents the number of people living in the region. Each part is a *district*. A larger geographic region (such as a state) must be partitioned into k districts when the state is to be represented in a legislative body by k people; each district elects a single representative. The partition is called a *districting plan*.

The rules governing this partitioning vary from place to place, but usually there are (at least) three important goals: *contiguity*, *population balance*, and *compactness*.[1]

- *Contiguity* is often interpreted as connectivity; we represent this by requiring that the set of small regions forming each district is connected via shared boundary edges.
- *Population balance* requires that two different districts have approximately equal numbers of people.
- One measure of *compactness* that has been advocated e.g. by DeFord, Duchin, Solomon, and Tenner [7, 8, 11, 12] is the number of pairs of adjacent small regions that lie in distinct districts, equivalent to the cardinality of the *cut-set* corrresponding to the partition.

Thus in the definitions of the *optimization* and *sampling* problems above, the connectivity constraint reflects the contiguity requirement, the part-weight constraint reflects the population balance requirement, and the cost is a measure of compactness.

The *optimization* problem described above arises in computer-assisted redistricting; an algorithm for solving this problem could be used to select a districting plan that is optimally compact subject to contiguity and desired population balance, where compactness is measured as discussed above.

The *sampling* problem arises in evaluating a plan; in court cases [4, 35, 24, 23, 36] expert witnesses argue that a districting plan reflects an intention to gerrymander by comparing it to districting plans randomly sampled from a distribution. The expert witnesses use Markov Chain Monte Carlo (MCMC), albeit unfortunately on Markov chains that have not been shown to be rapidly mixing, which means that the samples are possibly not chosen according to anything even close to a uniform distribution. There have been many papers addressing random sampling of districting plans (e.g. [1, 3, 8, 23, 24]) but, despite the important role of random sampling in court cases, there are no results on provably uniform or nearly uniform sampling from a set of realistic districting plans for a realistic input in a reasonable amount of time.

It is known that even basic versions of these problems are NP-hard. If the vertex weights are allowed to very large integers, expressed in binary, the NP-hardness of SUBSET SUM already implies the NP-completeness of partitioning the vertices into two equal-weight subsets. However, in application to redistricting the integers are not very large. For the purpose of seeking hardness results, it is better to focus on a special case, the *unit-weight* case, in which each vertex has weight one. For this case, Dyer and Frieze [13] showed that, for any fixed $p \geq 3$, it is NP-hard to find a weight-balanced partition of the vertices of a planar graph into connected parts of size p. Najt, Deford, and Solomon [33] showed that even without the constraint on balance, uniform sampling of partitions into *two* connected parts is NP-hard.

Following Ito et al. [27, 26] and Najt et al. [33], we therefore consider a further restriction on the input graph: we consider graphs with bounded *branchwidth/treewidth*.[2]

[1] These terms are often not formally defined in law.

[2] *Treewidth* and branchwidth are very similar measures; they are always within a small constant factor of each other. Thus a graph has small treewidth if and only if it has small branchwidth.

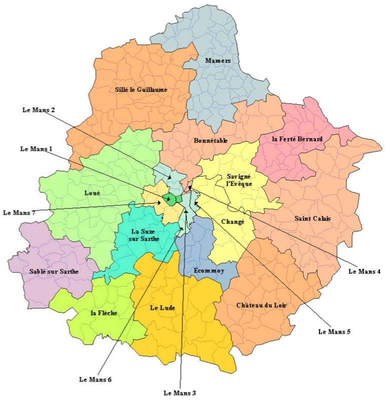


Figure 3 This map shows the twenty-one cantons for the department “Sarthe” of France. The cantons are the atomic regions for the redistricting of Sarthe. The corresponding radial graph has radius six, so there is a branch decomposition of width $w = 6$. For the upcoming redistricting of France, Sarthe must be divided into $k = 3$ districts.

The branchwidth of a graph is a measure of how treelike the graph is: often even an NP-hard graph problem is quickly solvable when the input is restricted to graphs with low branchwidth. For planar graphs in particular, there are known bounds on branchwidth that are relevant to the application. A planar graph on n vertices has branchwidth $O(\sqrt{n})$, and a planar graph of diameter d has branchwidth $O(d)$. There is an stronger bound, which we will review in Section 2.2.

Najt, Deford, and Solomon [33] show that, for any fixed k and fixed w, the optimization and sampling problems *without the constraint on population balance* can be solved in polynomial time on graphs of branchwidth at most w.[3] Significantly, the running time is of the form $O(f(k, w)n^c)$ for some constant c. Such an algorithm is said to be *fixed-parameter tractable* with respect to k and w, meaning that as long as k and w are fixed, the problem is considered tractable. Fixed-parameter tractability is an important and recognized way of coping with NP-completeness.

However, their result has two disadvantages. First, as the authors point out, the big O hides a constant that is astronomical; for NP-hard problems, one expect that the dependence on the parameters be at least exponential but in this case it is a tower of exponentials. As the authors state, the constants in the theorems on which they rely are “too large to be practically useful.”

Second, because their algorithm cannot handle the constraint on population balance, the algorithm would not be applicable to redistricting even if it were tractable. The authors discuss (Remark 5.11 in [33]) the extension of their approach to handle balance: “It is easy to add a relational formula...that restricts our count to only balanced connected k-partitions.... From this it should follow that ... [the problems are tractable]. However ... the corresponding meta-theorem appears to be missing from the literature.”

In our first result, we show that in fact what they seek does not exist: under a standard complexity-theoretic assumption, **there is no algorithm that is fixed-parameter tractable with respect to both k and w.**

More precisely, we use the analogue of NP-hardness for fixed-parameter tractability, $W[1]$-hardness. We show the following in Section 4.

[3] They use treewidth but the results are equivalent.

▶ **Theorem 1.** *For unit weights, finding a weight-balanced k-partition of a planar graph of width w into connected parts is $W[1]$-hard with respect to $k + w$.*

In the theory of fixed-parameter tractability (see e.g. Section 13.4 of [6]) this is strong evidence that no algorithm exists with a running time of the form $O(f(k, w)n^c)$ for fixed c independent of k and w.

This is bad news but there is a silver lining. The lower bound guides us in seeking good algorithms, and it does not rule out an algorithm that has a running time of the form $f(k)n^{O(w)}$ or $f(w)n^{O(k)}$. That is, according to the theory, while there is no algorithm that is fixed-parameter tractable with respect to both k and w simultaneously, there *could* be one that is fixed-parameter tractable with respect to k alone and one that is fixed-parameter tractable with respect to w alone.

These turn out to be true. First we discuss fixed-parameter tractability with respect to k. Ito et al. [27, 26] show that, even for general (not necessarily planar) graphs there is an algorithm with running time $O((w + 1)^{2(w+1)}U^{2(w+1)}k^2n)$, where U is the upper bound on the part weights. Thus for unit weights, the running time is $O((w + 1)^{2(w+1)}n^{2w+3})$.

However, for the application we have in mind this is not the bound to try for. Indeed, the motivation for this project arose from a collaboration between the first author and some other researchers. That team, in anticipation of the upcoming redistricting in France, sought to find good district plans with respect to various criteria for French departments. Their approach was to develop code that, for each department, would explicitly enumerate all district plans that (a) are connected and (b) are population-balanced to within 20% of the mean. Their effort succeeded on all but three departments (not including Paris, which follows different rules): Doubs (25), Saône-et-Loire (71) and, Seine-Maritime (76). The question arose: could another algorithmic approach succeed in finding optimal district plans for these under some objective function? We observed that the numbers of districts tend to be *very* small (sixty-three out of about a hundred departments have between two and five districts, and the average is a little over three.) The number of atoms of course tends to be much larger, but the diameter of the graph is often not so large, and hence the same is true for branchwidth.[4]

Thus, to address such instances, we need an algorithm that can tolerate a very small number k of districts and a moderately small branchwidth w. We prove the following in Section 5.

▶ **Theorem 2.** *For the optimization problem and the sampling problem, there are algorithms that run in $O(c^wU^kSn(\log U + \log S))$ time, where c is a constant, k is the number of districts, $w \geq k$ is an upper bound on the branchwidth of the planar graph, n is the number of vertices of the graph, U is the upper bound on the weight of a part, and S is an upper bound on the cost of a desired solution.*

Remarks.

1. In the unit-cost case (every edge cost is one), $S \leq n$.
2. In the unit-weight, unit-cost case, the running time is $O(c^wn^{k+2} \log n)$.
3. For practical use the input weights need not be the populations of the atoms; if approximate population is acceptable, the weight of an atom with population p can be, e.g., $\lceil p/1000 \rceil$.

[4] For example, the French redistricting instances all have branchwidth at most eight; the average is about five.

In order to demonstrate that the theoretical algorithm is not inherently impractical, we developed an implementation for the optimization problem, and successfully applied it to find solutions for the redistricting instances in France. French law requires that the population of each department needs to be within 20% of the mean. The implementation found the cut-size-minimizing solutions subject to the 20% population balance constraint, and subject to a 10% population balance constraint. Using a 5% population balance constraint, we found optimal solutions for over half of the departments. We briefly describe the results in Section 6, and we illustrate some district plans in the full version of the paper.

2 Preliminaries

2.1 Branchwidth

A *branch decomposition* of a graph G is a rooted binary tree with the following properties:

1. Each node x is labeled with a subset $C(x)$ of the edges of G.
2. The leaves correspond to the edges of G: for each edge e, there is a leaf x such that $C(x) = \{e\}$.
3. For each node x with children x_1 and x_2, $C(x)$ is the disjoint union of $C(x_1)$ and $C(x_2)$.

We refer to a set $C(x)$ as a *branch cluster*. A vertex v of G is a *boundary vertex* of $C(x)$ if G has at least one edge incident to v that is in $C(x)$ and at least one edge incident to v that is not in $C(x)$. The *width* of a branch cluster is the number of boundary vertices, and the width of a branch decomposition is the maximum cluster width. The branchwidth of a graph is the minimum w such that the graph has a branch decomposition of width w.

For many optimization problems in graphs, if the input graph is required to have small branchwidth then there is a fast algorithm, often linear time or nearly linear time, and often this algorithm can be adapted to do uniform random sampling of solutions. Therefore Najt, Deford, and Solomon [33] had good reason to expect that there would be a polynomial-time algorithm to sample from balanced partitions where the degree of the polynomial was independent of w and k.

2.2 Radial graph

For a planar embedded graph G, the radial graph of G has a node for every vertex of G and a node for every face of G, and an edge between a vertex-node and a face-node if the vertex lies on the face's boundary. Note that the radial graph of G is isomorphic to the radial graph of the dual of G. There is a linear-time algorithm that, given a planar embedded graph G and a node r of the radial graph, returns a branch decomposition whose width is at most the number of hops required to reach every node of the radial graph from r (see, e.g., [30]). For example, Figure 2 shows that the number of hops required is at most six, so the linear-time algorithm would return a branch decomposition of width w at most six.

Using this result, some real-world redistricting graphs can be shown to have moderately small branchwidth. For example, Figure 3 shows a department of France, Sarthe, that will need to be divided into $k = 3$ districts. The number of hops required for this example is six, so we would get a branch decomposition of width w at most six.

2.3 Sphere-cut decomposition

The branch decomposition of a planar embedded graph can be assumed to have a special form. The radial graph of G can be drawn on top of the embedding of G so that a face-node is embedded in the interior of a face of G and a vertex-node is embedded in the same location as the corresponding vertex. We can assume that the branch decomposition has the property that corresponding to each branch cluster C is a cycle in the radial graph that encloses exactly the edges belonging to the cluster C, and the vertices on the boundary of this cluster are the vertex-nodes on the cycle. This is called a *sphere-cut decomposition* [10]. If the branch decomposition is derived from the radial graph using the linear-time algorithm mentioned above, the sphere-cut decomposition comes for free. Otherwise, there is an $O(n^3)$ algorithm to find a given planar graph's least-width branch decomposition, and if this algorithm is used it again gives a sphere-cut decomposition.

3 Related work

There is a vast literature on partitioning graphs, in particular on partitions that are in a sense balanced. In particular, in the area of decomposition of planar graphs, there are algorithms [37, 34, 38] for *sparsest cut* and *quotient cut*, in which the goal is essentially to break off a single piece such that the cost of the cut is small compared to the amount of weight on the smaller side. The single piece can be required to be connected. There are approximation algorithms for variants of balanced partition [19, 17] into two pieces. These only address partitioning into $k = 2$ pieces, the pieces are not necessarily connected, and the balance constraint is only approximately satisfied. In one paper [29], the authors use a variant of binary decision diagrams to construct a compact representation of all partitions of a graph into k connected parts subject to a balance constraint. However, their algorithm does not address the problem of minimizing the size of the cut-set.

There are many papers on algorithms relevant to computer-aided redistricting (a few examples are [5, 14, 22, 25, 32, 18]). Note that in this paper we focus on algorithms that have guaranteed polynomial running times (with respect to fixed parameters k and w) and that are guaranteed to find optimal solutions or that provably generate random solutions according to the uniform distribution. There has been much work on using Markov Chain Monte Carlo as a heuristic for optimization or for random generation but so far such methods are not accompanied by mathematical guarantees as to running time or quality of output.

Finally, there many papers on $W[1]$-hardness and more generally lower bounds on fixed-parameter tractability, as this is a well-studied area of theoretical computer science. Our result is somewhat rare in that most graph problems are fixed-parameter tractable with respect to branchwidth/treewidth. However, there are by now other $W[1]$-hardness results with respect to treewidth [9, 2, 16, 31, 21, 20] and a few results [2, 15] were previously known even under the restriction that the input graph must be planar.

4 W[1]-Hardness

In this section, we show that the problem is W[1]-hard parameterized by $k + w$, where k is the number of districts and w the treewidth of the graph.

We start with the following lemma that shows that it is enough to prove that a more structured version of the problem (bounded vertex weights, each region must have size greater than 1) is W[1]-hard.

▶ **Lemma 3.** *If the planar vertex-weighted version of the problem is W[1]-hard parameterized by $k + w$ when the total weight of each region should be greater than 1, and the smallest weight is 1 and the largest weight is polynomial in the input size, then the planar unweighted version of the problem is W[1]-hard parameterized by $k + w$.*

Proof. Consider a weighted instance of the problem satisfying the hypothesis of the lemma. Let $w_{\min}$ and $W_{\max}$ respectively denote the minimum and maximum weights. First, rescale all the weights of the vertices so as to make them integers. Since the input weights are rationals and $W_{\max}$ is polynomial in the input size, this does not change the size complexity of the problem by more than a polynomial factor. We now make the following transformations to the instance. For each vertex v of weight $w(v)$, create $w(v)-1$ unit-weight *dummy* vertices and connect each of them to v with a single edge, then remove the weight of v.

This yields a unit-weight graph which satisfies the following properties. First, if the input graph was planar, then the resulting graph is also planar. Second, since the ratio $W_{\max}$ is polynomial in the input size, the total number of vertices in the new graph is polynomial in the input size. Finally, any solution for the problem on the vertex-weighted graph can be associated to a solution for the problem on the unit-weight graph: for each vertex v of the original graph, assign each of the $w(v) - 1$ dummy vertices to the same region as v. We have that the associated solution has connected regions of exactly the same weight as the solution in the weighted graph. Moreover, we claim that any solution for the unit-weight graph is associated to a solution of the input weighted graph: this follows from the assumption that the prescribed weights for the regions is greater than 1 and that the regions must be connected. Thus for each vertex v, in any solution all the $w(v) - 1$ dummy vertices must belong to the region of v.

Therefore, if the planar vertex-weighted version of the problem is W[1]-hard parameterized by $k + w$ when the smallest weight is at least 1, the total weight of each region should be greater than 1, and the sum of the vertex weights of the graph is polynomial in the input size, then the planar unit-weight version of the problem is W[1]-hard parameterized by $k + w$. ◀

By Lemma 3, we can focus without loss of generality on instances $G = (V, E), w : V \mapsto \mathbb{R}_+$ where the vertex weights w lie in the interval $[1, |V|^c]$ for some absolute constant c. We next show that the problem is W[1]-hard on these instances.

We reduce from the Bin Packing problem with polynomial weights. Given a set of integer values $v_1, \ldots, v_n$ and two integers B and k, the *Bin Packing* problem asks to decide whether there exists a partition of $v_1, \ldots, v_n$ into k parts such that for each part of the partition, the sum of the values is at most B. The Bin Packing problem with polynomially bounded weights assumes that there exists a constant c such that $B = O(n^c)$. Note that for the case where the weights are polynomially bounded, we can assume w.l.o.g. that the sum of the weights is exactly kB by adding $kB - \sum_{i=1}^{n} v_i$ elements of value 1. Since the weights are polynomially bounded and that each weight is integer we have that (1) the total number of new elements added is polynomial in n, hence the size of the problem is polynomial in n, and (2) there is a solution to the original problem if and only if there is a solution to the new problem: the new elements can be added to fill up the bins that are not full in the solution of the original problem.

We will make use of the following theorem of Jansen et al. [28].

▶ **Theorem 4** ([28]). *The Bin Packing problem with polynomial weights is W[1]-hard parameterized by the number of bins k. Moreover, there is no $f(k)n^{o(k/\log k)}$ time algorithm assuming the exponential time hypothesis (ETH).*

We now proceed to the proof of Theorem 1. From an instance of Bin Packing with polynomially bounded weights and whose sum of weights is kB, create the following instance for the problem. For each $i \in [2n + 1]$, create

$$\ell_i = \begin{cases} k & \text{if } i \text{ is odd} \\ k+1 & \text{if } i \text{ is even} \end{cases}$$

vertices $s_i^1, \ldots, s_i^{\ell_i}$. Let $S_i = \{s_i^1, \ldots, s_i^{\ell(i)}\}$. Moreover, for each odd $i < n$, for each $1 \le j \le k$, connect s_i^j to s_{i-1}^j and s_{i+1}^j, and when $j < k$, also to s_{i-1}^{j+1} and s_{i+1}^{j+1}. Let G be the resulting graph.

It is easy to see that G is planar. We let f_∞ be the longest face: $\{s_1^1, \ldots, s_1^k, s_2^{k+1}, s_3^k, \ldots, s_{2n+1}^k, s_{2n+1}^{k-1}, \ldots, s_{2n+1}^1, s_{2n}^1, \ldots, s_2^1\}$. We claim that the treewidth of the graph is at most $7k$. To show this we argue that the face-vertex incidence graph $\bar{G}$ of G has diameter at most $2k+4$ and by Lemma 3 this immediately yields that the treewidth of G is at most $10k$. We show that each vertex of $\bar{G}$ is at hop-distance at most $k+2$ of the vertex corresponding to f_∞. Indeed, consider a vertex s_i^j (for a face, consider a vertex s_i^j on that face). Recall that for each i_0, j_0, we have that $s_{i_0}^{j_0}$ is adjacent to $s_{i+1}^{j_0}$ and $s_{i+1}^{j_0+1}$ and so, s_i^j is at hop-distance at most $k+1$ from either $s_i^{\ell(i)}$ or s_i^1 in $\bar{G}$. Moreover both s_i^1 and $s_n^{\ell(i)}$ are on face f_∞ and so s_i^j is at hop-distance at most $k+2$ from f_∞ in $\bar{G}$. Hence the treewidth of G is at most $10k$.

Our next step is to assign weights to the vertices. Then, we set the weight $w(s_i^j)$ of every vertex s_i^j of $\{s_1^1, \ldots, s_1^k\}$ to be $(kB)^2$ and the weight $w(s_i^j)$ of every vertex s_i^j of $\{s_{2n+1}^1, \ldots, s_{2n+1}^k\}$ to be $(kB)^4$. For each odd $i \neq 1, 2n+1$ we set a weight of $1/(2n-2)$. Finally, we set the weight of each vertex s_i^j where i is even to be $v_{i/2}$. Let $T = (kB)^2 + (kB)^4 + 1/2 + kB$, and recall that $kB = \sum_{i=1}^n v_i$.

▶ **Fact 1.** *Consider a set S of vertices containing exactly one vertex of S_i for each i. Then the sum of the weights of the vertices in S is T.*

We now make the target weight of each region to be $(kB)^2 + (kB)^4 + kB + B = T + B$. We have the following lemma.

▶ **Lemma 5.** *In any feasible solution to the problem, there is exactly 1 vertex of $\{s_1^1, \ldots, s_1^k\}$ and exactly 1 vertex of $\{s_n^1, \ldots, s_n^{\ell(n)}\}$ in each region.*

Proof. Recall that by definition we have that $\sum_{i=1}^n v_i = kB$. Moreover, the number of vertices with weight equal to $(kB)^2$ is exactly k. Thus, since the target weight of each region is $(kB)^2 + (kB)^4 + B + kB$, each region has to contain exactly 1 vertex from $\{s_1^1, \ldots, s_1^k\}$ and exactly 1 vertex from $\{s_n^1, \ldots, s_n^{\ell(n)}\}$. ◀

We now turn to the proof of completeness and soundness of the reduction. We first show that if there exists a solution to the Bin Packing instance, namely that there is a partition into k parts such that for each part of the partition, the sum of the values is B, then there exists a feasible solution to the problem. Indeed, consider a solution to the Bin Packing instance $\{B_1, \ldots, B_k\}$ and construct the following solution to the problem. For each odd i, assign vertices $s_i^1, \ldots, s_i^k$ to regions $R_1, \ldots, R_k$ respectively. For each $i \in [n]$, perform the following assignment for the even rows. Let u_i be the integer in $[k]$ such that $v_i \in B_{u_i}$. Assign all vertices $s_{2i}^1, \ldots, s_{2i}^{u_i-1}$ to regions $R_1, \ldots R_{u_i-1}$ respectively. Assign both vertices $s_{2i}^{u_i}$ and $s_{2i}^{u_i+1}$ to region R_{u_i}. Assign all vertices $s_{2i}^{u_i+2}, \ldots s_{2i}^{k+1}$ to regions $R_{u_i+1}, \ldots R_k$. The connectivity of the regions follows from the fact that for each odd i, s_i^j is connected to both s_{i+1}^j and s_{i+1}^{j+1} and to both s_{i-1}^j and s_{i-1}^{j+1}.

We then bound the total weight of each region. Let's partition the vertices of a region R_j into two: Let S_{R_j} be a set that contains one vertex from each S_i and let $\bar{S}_{R_j}$ be the rest of the elements. The total weight of the vertices in S_{R_j} is by Fact 1 exactly T. The total weight of the remaining vertices corresponds to the sum of the values v_i such that $|R_j \cap S_i| = 2$ which is $\sum_{v_i \in B_j} v_i = B$ since it is a solution to the Bin Packing problem. Hence the total weight of the region is $T + B$, as prescribed by the problem.

We finally prove that if there exists a solution for the problem with the prescribed region weights, then there exists a solution to the Bin Packing problem. Let $R_1, \ldots, R_k$ be the solution to the problem. By Lemma 5, each region contains one vertex of $s_1^1, \ldots s_1^k$ and one vertex of $s_1^1, \ldots s_{2n+1}^k$. Since the regions are required to be connected, there exists a path joining these two vertices and so by the pigeonhole principle for each odd i, each region contains exactly one vertex of $s_i^1, \ldots s_i^k$. Moreover for each even i, each region contains at least one vertex of $s_i^1, \ldots s_i^{k+1}$ and exactly one region contains two vertices. Let $\phi(i) \in [k]$ be such that $|R_{\phi(i)} \cap \{s_i^1, \ldots s_i^{k+1}\}| = 2$. We now define the following solution for the Bin Packing problem. Define the jth bin as $B_j = \{v_i \mid \phi(i) = j\}$. We claim that for each bin B_j the sum of the weights of the elements in B_j is exactly B. Indeed, observe that region R_j contains exactly one vertex of $s_i^1, \ldots s_i^k$ for each odd i and exactly one vertex of $s_i^1, \ldots s_i^{k+1}$ for each even i except for the sets $s_i^1, \ldots s_i^{k+1}$ where $\phi(i) = j$ for which it contains two vertices. Thus by Fact 1, the total sum of the weights is $T + \sum_{i|\phi(i)=j} v_i$ and since the target weight is $T + B$ we have that $\sum_{i|\phi(i)=j} v_i = B$. Since the weight of B_j is exactly $\sum_{i|\phi(i)=j} v_i$ the proof is complete.

5 Algorithm

In this section, we describe the algorithms of Theorem 2. In describing the algorithm, we will focus on simplicity rather than on achieving the best constant possible as the base of k.

5.1 Partitions

A *partition* of a finite set Ω is a collection of disjoint subsets of Ω whose union is Ω. A partition defines an equivalence relation on Ω: two elements are equivalent if they are in the same subset.

There is a partial order on partitions of Ω: $\pi_1 \prec \pi_2$ if every part of π_1 is a subset of a part of π_2. This partial order is a lattice. In particular, for any pair π_1, π_2 of partitions of Ω, there is a unique minimal partition π_3 such that $\pi_1 \prec \pi_3$ and $\pi_2 \prec \pi_3$. (By *minimal*, we mean that for any partition π_4 such that $\pi_1 \prec \pi_4$ and $\pi_2 \prec \pi_4$, it is the case that $\pi_3 \prec \pi_4$.) This unique minimal partition is called the *join* of π_1 and π_2, and is denoted $\pi_1 \vee \pi_2$.

It is easy to compute $\pi_1 \vee \pi_2$: initialize $\pi := \pi_1$, and then repeatedly merge parts that intersect a common part of π_2.

In a slight abuse of notation, we define the join of a partition π_1 of one finite set Ω_1 and a partition π_2 of another finite set Ω_2. The result, again written $\pi_1 \vee \pi_2$, is a partition of $\Omega_1 \cup \Omega_2$. It can be defined algorithmically: iniitalize π to consist of the parts of π_2, together with a singleton part $\{\omega\}$ for each $\omega \in \Omega_2 - \Omega_1$. Then repeatedly merge parts of π that intersect a common part of π_2.

5.2 Noncrossing partitions

The sphere-cut decomposition is algorithmically useful because it restricts the way a graph-theoretic structure (such as a solution) can interact with each cluster. For a cluster C, consider the corresponding cycle in the radial graph, and let θ_C be the cyclic permutation $(v_1\ v_2\ \cdots\ v_m)$ of boundary vertices in the order in which they appear in the radial cycle. (By a slight abuse of notation, we may also interpret θ_C as the set $\{v_1, \ldots, v_m\}$.

First consider a partition ρ^{in} of the vertices incident to edges belonging to C, with the property that each part induces a connected subgraph of C. Planarity implies that the partition induced by ρ^{in} on the boundary vertices $\{v_1, \dots, v_m\}$ has a special property.

▶ **Definition 6.** *Let π be a partition of the set $\{1, \dots, m\}$. We say π is* crossing *if there are integers $a < b < c < d$ such that one part contains a and c and another part contains b and d.*

It follows from connectivity that the partition induced by ρ^{in} on the boundary vertices θ_C is a noncrossing partition. Similarly, let ρ^{out} be a partition of the vertices incident to edges that do *not* belong to C; then ρ^{out} induces a noncrossing partition on the boundary vertices of C.

The asymptotics of the Catalan numbers imply the following (see, e.g., [10]).

▶ **Lemma 7.** *There is a constant c_1 such that the number of noncrossing partitions of $\{1, \dots, w\}$ is $O(c_1^w)$.*

Finally, suppose ρ is a partition of all vertices of G such that each part is connected. Then $\rho = \rho^{\text{in}} \vee \rho^{\text{out}}$ where ρ^{in} is a partition of the vertices incident to edges in C (in which each part is connected) and ρ^{out} is a partition of the vertices incident to edges not in C (in which each part is connected).

Because the only vertices in both ρ^{in} and ρ^{out} are those in θ_C, the partition ρ induces on θ_C is $\pi^{\text{in}} \vee \pi^{\text{out}}$ where π^{in} is the partition induced on θ_C by ρ^{in} and π^{out} is the partition induced on θ_C by ρ^{out}.

5.3 Algorithm overview

The algorithms for optimization and sampling are closely related.

The algorithms are based on dynamic programming using the sphere-cut decomposition of the planar embedded input graph G.

Each algorithm considers every vertex v of the input graph and selects one edge e that is incident to v, and designates each branch cluster that contains e as a *home cluster* for v.

We define a *topological configuration* of a cluster C to be a pair $(\pi^{\text{in}}, \pi^{\text{out}})$ of noncrossing partitions of θ_C with the following property:

$$\pi^{\text{in}} \vee \pi^{\text{out}} \text{ has at most } k \text{ parts.} \tag{1}$$

The intended interpretation is that there exist ρ^{in} and ρ^{out} as defined in Section 5.2 such that ϕ^{in} is the partition ρ^{in} induces on θ_C and ϕ^{out} is the partition ρ^{out} induces on θ_C.

We can assume that the vertices of the graph are assigned unique integer IDs, and that therefore there is a fixed total ordering of θ_C. Based on this total ordering, for any partition π of θ_C, let p be the number of parts of π, and define representatives(π) to be the p-vector $(v_1, v_2, \dots, v_p)$ obtained as follows:

- v_1 is the smallest-ID vertex in θ_C,
- v_2 is the smallest-ID vertex in θ_C that is not in the same part as v_1,
- v_2 is the smallest-ID vertex in θ_C that is not in the same part as v_1 and is not in the same part as v_2,

and so on.

This induces a fixed total ordering of the parts of $\pi^{\text{in}} \vee \pi^{\text{out}}$.

We define a *weight configuration* of C to be a k-vector $\boldsymbol{w} = (w_1, \dots, w_k)$ where each w_i is a nonnegative integer less than U. There are U^k such vectors.

We define a *weight/cost configuration* of C to be a k-vector together with a nonnegative integer s less than S. There are $U^k S$ such configurations.

We define a *configuration* of C to be a pair consisting of a topological configuration and a weight/cost configuration. The number of configurations of C is bounded by $c^w U^k S$.

The algorithms use dynamic programming to construct, for each cluster C, a table T_C indexed by configurations of C. In the case of optimization, the table entry $T_C[\Psi]$ corresponding to a configuration Ψ is *true* or *false*. For sampling, $T_C[\Psi]$ is a cardinality.

Let $\Psi = ((\pi^{\text{in}}, \pi^{\text{out}}), ((w_1, \ldots, w_k), s))$ be a configuration of C. Let $\text{count}(\Psi)$ be the number of partitions ρ^{in} of the vertices incident to edges belonging to C with the following properties:

- ρ^{in} induces π^{in} on θ_C.
- Let $\pi = \pi^{\text{in}} \vee \phi^{\text{out}}$. Let $\text{representatives}(\pi) = (v_1, \ldots, v_p)$. Then for $j = 1, \ldots, p$, w_j is the total weight of vertices v for which C is a home cluster and such that v belongs to the same part of $\rho^{\text{in}} \vee \pi^{\text{out}}$ as v_j.

For optimization, $T_C[\Psi]$ is true if $\text{count}(\Psi)$ is nonzero. For sampling, $T_C[\Psi] = \text{count}(\Psi)$. We describe in Section 5.5 how to populate these tables. Next we describe how they can be used to solve the problems.

5.4 Using the tables

For the root cluster $\hat{C}$, the cluster that contains all edges of G, $\theta_{\hat{C}}$ is empty. Therefore there is only one partition of $\theta_{\hat{C}}$, the trivial partition π_0 consisting of a single part, the empty set.

To detemine the optimum cost in the optimization problem, simply find the minimum nonnegative integer s such that, for some $\boldsymbol{w} = (w_1, \ldots, w_k)$ such that each w_i lies in $[L, U)$, the entry $T_{\hat{C}}[((\pi_0, \pi_0), (\boldsymbol{w}, s))]$ is *true*. To find the solution with this cost, the algorithm needs to find a "corresponding" configuration for each leaf cluster $C(\{uv\})$; that configuration tells the algorithm whether the two endpoints u and v are in the same district. This information is obtained by a recursive algorithm, which we presently describe.

Let C_0 be a cluster with child clusters C_1 and C_2. For $i = 0, 1, 2$, let $(\pi_i^{\text{in}}, \pi_i^{\text{out}})$ be a topological configuration for cluster C_i. Then we say these topological configurations are *consistent* if the following properties hold:

- For $i = 1, 2$, $\pi_i^{\text{out}} = \pi_0^{\text{out}} \vee \pi_{3-i}^{\text{in}}$.
- $\pi_0^{\text{in}} = \pi_1^{\text{in}} \vee \pi_2^{\text{in}}$.

For $i = 0, 1, 2$, let $(\boldsymbol{w}_i, s_i)$ be a weight/cost configuration for C_i. We say they are consistent if $\boldsymbol{w}_0 = \boldsymbol{w}_1 + \boldsymbol{w}_2$ and $s_0 = s_1 + s_2$.

Finally, for $i = 0, 1, 2$, let $\Psi_i = ((\pi_i^{\text{in}}, \pi_i^{\text{out}}), (\boldsymbol{w}_i, s_i))$ be a configuration for cluster C_i. Then we say Ψ_1, Ψ_2, Ψ_3 are consistent if the topological configurations are consistent and the weight/cost configurations are consistent.

▶ **Lemma 8.** *For a configuration Ψ_0 of C_0, count(Ψ_0) $= \sum_{\Psi_1, \Psi_2}$ count(Ψ_1) $\cdot$ count(Ψ_2) where the sum is over pairs (Ψ_1, Ψ_2) of configurations of C_1, C_2 such that Ψ_0, Ψ_1, Ψ_2 are consistent.*

The recursive algorithm, given a configuration Ψ for a cluster C such that $T_C[\Psi]$ is *true*, finds configurations for all the clusters that are descendants of C such that, for each nonleaf descendant and its children, the corresponding configurations are consistent; for each descendant cluster C', the configuration Ψ' selected for it must have the property that $T_{C'}[\Psi']$ is *true*.

The algorithm is straightforward:

Algorithm 1 Descend(C_0, Ψ_0).

```
define Descend($C_0, \Psi_0$):
 precondition: $T_{C_0}[\Psi_0] = true$
assign $\Psi_0$ to $C_0$
 if $C_0$ is not a leaf config
   for each config $\Psi_1 = ((\pi_1^{\text{in}}, \pi_1^{\text{out}}), (\boldsymbol{w}_1, s_1))$ of $C_0$'s left child $C_1$,
     if $T_{C_1}[\Psi_1]$ is true
       for each topological config $(\pi_2^{\text{in}}, \pi_2^{\text{out}})$ of $C_0$'s right child $C_2$
         let $(\boldsymbol{w}_2, s_2)$ be the weight/cost config of $C_2$ such that
           $\Psi_0, \Psi_1, \Psi_2$ are consistent
             where $\Psi_2 = ((\pi_2^{\text{in}}, \pi_2^{\text{out}}), (\boldsymbol{w}_2, s_2))$
         if $T_{C_2}[\Psi_2] = true$
           call Descend($C_1, \Psi_1$) and Descend($C_2, \Psi_2$)
           return, exiting out of loops
```

Lemma 8 shows via induction from root to leaves that the procedure will successfully find configurations for all clusters that are descendants of C_0. For the root cluster $\hat{C}$ and a configuration $\hat{\Psi}$ of $\hat{C}$ such that $T_{\hat{C}}[\hat{\Psi}]$ is *true*, consider the Ψ_C configurations found for each leaf cluster, and let $(\pi_C^{\text{in}}, \pi_C^{\text{out}})$ be the topological configuration of Ψ_C Consider the partition

$$\rho = \bigvee_C \pi_C^{\text{in}}$$

where the join is over all leaf clusters C. Because there are no vertices of degree one, for each leaf cluster $C(\{uv\})$, both u and v are boundary vertices, so ρ is a partition of all vertices of the input graph. Induction from leaves to root shows that this partition agrees with the weight/cost part $(\hat{\boldsymbol{w}}, \hat{s})$ of the configuration $\hat{\Psi}$. In particular, the weights of the parts of ρ correspond to the weights of $\hat{w}$, and the cost of the partition equals $\hat{s}$.

In the step of Descend that selects $(\boldsymbol{w}_2, s_2)$, there is exactly one weight/cost config that is consistent (it can be obtained by permuting the elements of $\boldsymbol{w}_1$ and then subtracting from $\boldsymbol{w}_0$ and subtracting s_1 from s_0). By an appropriate choice of an indexing data structure to represent the tables, we can ensure that the running time of Descend is within the running time stated in Theorem 2. For optimization, it remains to show how to populate the tables.

Algorithm 2 Descend(C_0, Ψ_0, p).

```
define Descend($C_0, \Psi_0, p$):
 precondition: $p \leq T_{C_0}[\Psi_0]$
 assign $\Psi_0$ to $C_0$
 if $C_0$ is not a leaf config
   for each config $\Psi_1 = ((\pi_1^{\text{in}}, \pi_1^{\text{out}}), (\boldsymbol{w}_1, s_1))$ of $C_0$'s left child $C_1$,
     for each topological config $(\pi_2^{\text{in}}, \pi_2^{\text{out}})$ of $C_0$'s right child $C_2$
       let $(\boldsymbol{w}_2, s_2)$ be the weight/cost config of $C_2$ such that
         $\Psi_0, \Psi_1, \Psi_2$ are consistent
           where $\Psi_2 = ((\pi_2^{\text{in}}, \pi_2^{\text{out}}), (\boldsymbol{w}_2, s_2))$
       $\Delta := T_{C_1}[\Psi_1] \cdot T_{C_2}[\Psi_2]$
       if $p \leq \Delta$
         $q := \lfloor p / T_{C_2}[\Psi_2] \rfloor$
         $r := r \bmod T_{C_2}[\Psi_2]$
         call Descend($C_1, \Psi_1, q$) and Descend($C_2, \Psi_2, r$)
         return
       else $p := p - \Delta$ and continue
```

Induction shows that this procedure, applied to root cluster $\hat{C}$ and a configuration $\hat{\Psi}$ and an integer $p \leq T_{\hat{C}}[\hat{\Psi}]$, selects the p^{th} solution among those "compatible" with $\hat{\Psi}$. This can be used for random generation of solutions with given district populations and a given cost. Again, the running time for the procedure is within that stated in Theorem 2.

5.5 Populating the tables

For this section, let us focus on the tables needed for sampling. Populating the table for a leaf cluster is straightforward. Therefore, suppose C_0 is a cluster with children C_1 and C_2. We first observe that, given noncrossing partitions π_0^{out} of θ_{C_0}, π_1^{in} of θ_{C_1}, and π_2^{in} of θ_{C_2}, there are unique partitions $\pi_0^{\text{in}}, \pi_1^{\text{out}}, \pi_2^{\text{out}}$ such that the topological configurations $(\pi_0^{\text{in}}, \pi_0^{\text{out}}), (\pi_1^{\text{in}}, \pi_1^{\text{out}}), (\pi_2^{\text{in}}, \pi_2^{\text{out}})$ are consistent. (The formulas that show this are in the pseudocode below.)

The second observation: consider a configuration $\Psi_0 = (\kappa_0, (\boldsymbol{w}_0, s_0))$ of C_0. Then $\text{count}(\Psi_0)$ is

$$\sum_{\kappa_1,\kappa_2} \sum_{(\boldsymbol{w}_1,s_1),(\boldsymbol{w}_2,s_2)} \text{count}((\kappa_1, (\boldsymbol{w}_1, s_1))) \cdot \text{count}((\kappa_2, (\boldsymbol{w}_2, s_2))) \tag{2}$$

where the first sum is over pairs of topological configurations κ_1 for C_1 and and κ_2 where $\kappa_0, \kappa_1, \kappa_2$ are consistent, and the second sum is over pairs of weight/cost configurations that are consistent with $(\boldsymbol{w}_0, s_0)$. Note that because of how weight/cost configuration consistency is defined, the second sum mimics multivariate polynomial multiplication. We use these observations to define the procedure that populates the table for C_0 from the tables for C_1 and C_2.

Algorithm 3 COMBINE(C_0, C_1, C_2).

```
def COMBINE$(C_0, C_1, C_2)$:
 initialize each entry of $T_{C_0}$ to zero
 for each noncrossing partition $\pi_0^{\text{out}}$ of $\theta_{C_0}$
   for each noncrossing partition $\pi_1^{\text{in}}$ of $\theta_{C_1}$
     for each noncrossing partition $\pi_2^{\text{in}}$ of $\theta_{C_2}$
       $\pi_1^{\text{out}} = \pi_0^{\text{out}} \vee \pi_2^{\text{in}}$
       $\pi_2^{\text{out}} = \pi_0^{\text{out}} \vee \pi_1^{\text{in}}$
       $\pi_0^{\text{in}} = \pi_1^{\text{in}} \vee \pi_2^{\text{in}}$
       comment: now we populate entries of $T_{C_0}[\cdot]$ indexed by
                configurations of $C_0$ with
                topological configuration $(\pi_0^{\text{in}}, \pi_0^{\text{out}})$.
       for $i = 1, 2$,
         let $p_i(\boldsymbol{x}, y)$ be a polynomial over variables $x_1, \ldots, x_k, y$
           such that the coefficient of $x_1^{w_1} \cdots x_k^{w_k} y^s$
           is $T_{C_i}[((\pi_0^{\text{in}}, \pi_0^{\text{out}}), ((w_1, \ldots, w_k), s))]$
       let $p(\boldsymbol{x}, y)$ be the product of $p_1(\boldsymbol{x}, y)$ and $p_2(\boldsymbol{x}, y)$
       for every weight/cost configuration $((w_1, \ldots, w_k), s)$
         add to $T[((\pi_0^{\text{in}}, \pi_0^{\text{out}}), ((w_1, \ldots, w_k), s))]$ the
           coefficient of $x_1^{w_1} \cdots x_k^{w_k} y^s$ in $p(\boldsymbol{x}, y)$
```

The three loops involve at most c^w iterations, for some constant c. Multivariate polynomial multiplication can be done using multidimensional FFT. The time required is $O(N \log N)$, where $N = U^k S$. (This use of FFT to speed up an algorithm is by now a standard algorithmic technique.) It follows that the running time of the algorithm to populate the tables is as described in Theorem 2.

6 Implementation, and application to redistricting in France

Our implementation differs from the algorithm described in Section 5 in a few minor ways. Each configuration stores the populations of districts that intersect its boundary in a canonical order, as opposed to storing a k-vector containing the populations of all k districts. This reduces the number of configurations by reducing the redundancy of multiple configurations which are the same up to the ordering of the districts.

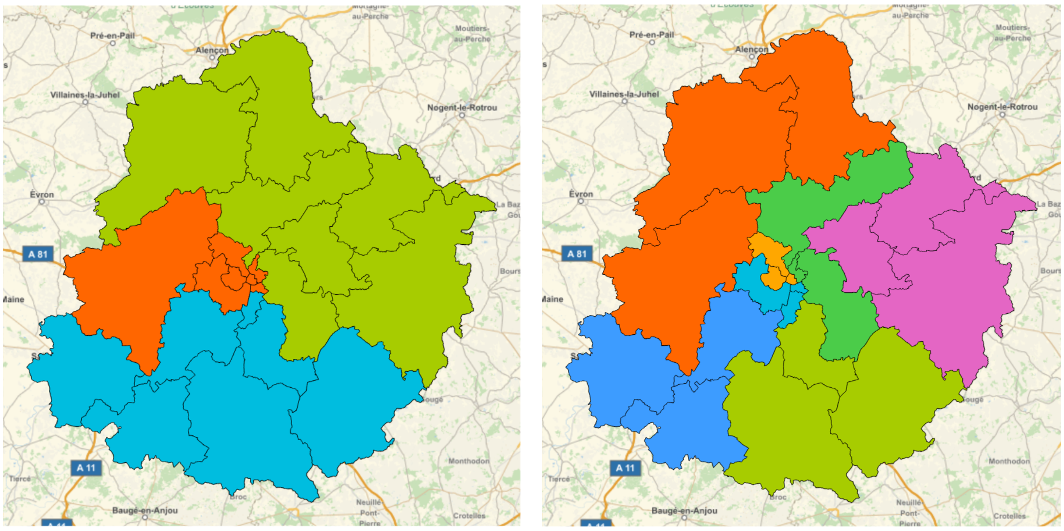
Also, our implementation does not use the FFT-based method for combining configurations; that method is helpful when the number of configurations is close to the maximum possible number but we expect that in practice the number will be substantially lower.

To demonstrate the effectiveness of our implementation, we applied it to the redistricting instances in France. There are about a hundred *departments* in France. The atoms are called *cantons*. For each department, one must find a partition of the cantons. Each part must be connected and each part's population can differ from the average by at most 20%. Omitting the special department of Paris (because its structure and rules are different) and the departments for which the target number of districts is one, we are left with eighty departments. The implementation was able to find solutions for every department. Additionally we were able to find solutions for over half of the departments with a tighter bound of 5%.

We were able to compute these solutions for all departments on a single machine within eight hours. As shown in Figure 5 the cut edge size of the optimal solution increases only slightly as the population constraint increases. This data suggests there is little downside to creating departments with closer populations when such a solution exists.

6.1 Example: *Sarthe*

Consider for example the department *Sarthe*. We specify that the minimum population of a district is 150,000 and the maximum population is 200,000. The computation took about 30 seconds on a single core of a 2018 MacBook Pro (Figure 4a).



(a) A districting of the cantons of Sarthe, France, generated with four districts.

(b) Sarthe with seven districts.

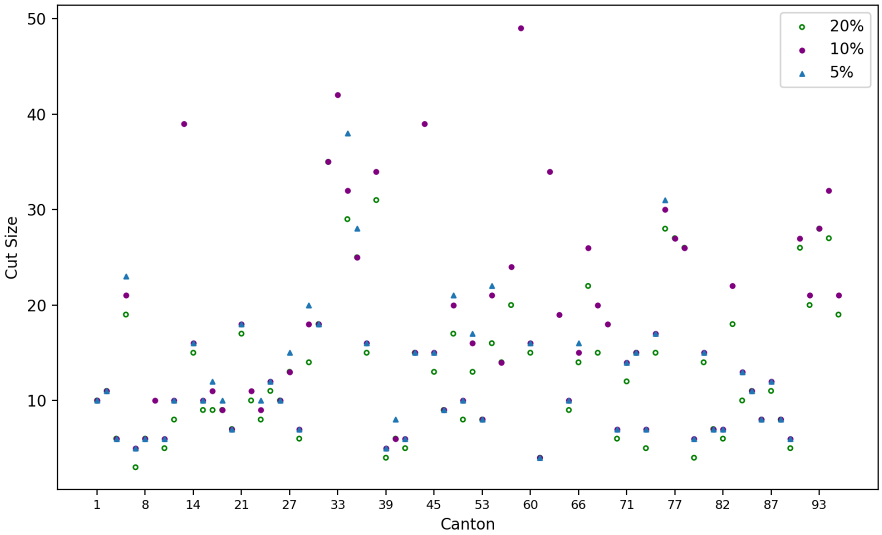


Figure 5 Differences in cut size cost for different population constraints. We include only those instances for which our implementation finds a solution.

References

1 Sachet Bangia, Christy Vaughn Graves, Gregory Herschlag, Han Sung Kang, Justin Luo, Jonathan C. Mattingly, and Robert Ravier. Redistricting: Drawing the line, 2017. `arXiv:1704.03360`.

2 Hans L. Bodlaender, Daniel Lokshtanov, and Eelko Penninkx. Planar capacitated dominating set is $W[1]$-hard. In Jianer Chen and Fedor V. Fomin, editors, *Proceedings of the 4th International Workshop on Parameterized and Exact Computation*, volume 5917 of *Lecture Notes in Computer Science*, pages 50–60. Springer, 2009. `doi:10.1007/978-3-642-11269-0_4`.

3 Daniel Carter, Gregory Herschlag, Zach Hunter, and Jonathan Mattingly. A merge-split proposal for reversible Monte Carlo Markov Chain sampling of redistricting plans, 2019. `arXiv:1911.01503`.

4 J. Chen. Expert report of Jowei Chen, Ph.D., Raleigh Wake Citizen's Association et al. vs. the Wake County Board of Elections, 2017. URL: `https://www.pubintlaw.org/wp-content/uploads/2017/06/Expert-Report-Jowei-Chen.pdf`.

5 Vincent Cohen-Addad, Philip N. Klein, and Neal E. Young. Balanced centroidal power diagrams for redistricting. In *Proceedings of the 26th ACM International Conference on Advances in Geographic Information Systems*, pages 389–396, 2018. `doi:10.1145/3274895.3274979`.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 1st edition, 2015.

7 Daryl DeFord and Moon Duchin. Redistricting reform in Virginia: Districting criteria in context. *Virginia Policy Review*, 12(2):120–146, 2019.

8 Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of Markov chains for redistricting, 2019. `arXiv:1911.05725`.

9 Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Capacitated domination and covering: A parameterized perspective. In *Proceedings of the 3rd International WorkshopParameterized and Exact Computation*, volume 5018 of *Lecture Notes in Computer Science*, pages 78–90. Springer, 2008. `doi:10.1007/978-3-540-79723-4_9`.

10 Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2010. doi:10.1007/s00453-009-9296-1.

11 Moon Duchin. Geography meets geometry in redistricting. Conference at Center for Geographic Analysis at Harvard University, May 2019. URL: https://cga-download.hmdc.harvard.edu/publish_web/CGA_Conferences/2019_Redistricting/slides/Moon_Duchin.pdf.

12 Moon Duchin and Bridget Eileen Tenner. Discrete geometry for electoral geography, 2018. arXiv:1808.05860.

13 Martin E. Dyer and Alan M. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discret. Appl. Math.*, 10(2):139–153, 1985. doi:10.1016/0166-218X(85)90008-3.

14 David Eppstein, Michael T. Goodrich, Doruk Korkmaz, and Nil Mamano. Defining equitable geographic districts in road networks via stable matching. In *Proceedings of the 25th ACM International Conference on Advances in Geographic Information Systems*, pages 52:1–52:4, 2017. doi:10.1145/3139958.3140015.

15 Andreas Emil Feldmann and Dániel Marx. The parameterized hardness of the k-center problem in transportation networks. *Algorithmica*, 82(7):1989–2005, 2020. doi:10.1007/s00453-020-00683-w.

16 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Inf. Comput.*, 209(2):143–153, 2011.

17 Kyle Fox, Philip N. Klein, and Shay Mozes. A polynomial-time bicriteria approximation scheme for planar bisection. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 841–850, 2015. doi:10.1145/2746539.2746564.

18 R. S. Garfinkel and G. L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B–495, 1970.

19 Naveen Garg, Huzur Saran, and Vijay V. Vazirani. Finding separator cuts in planar graphs within twice the optimal. *SIAM J. Comput.*, 29(1):159–179, 1999. doi:10.1137/S0097539794271692.

20 Sushmita Gupta, Saket Saurabh, and Meirav Zehavi. On treewidth and stable marriage. *CoRR*, abs/1707.05404, 2017. arXiv:1707.05404.

21 Gregory Z. Gutin, Mark Jones, and Magnus Wahlström. The mixed Chinese postman problem parameterized by pathwidth and treedepth. *SIAM J. Discret. Math.*, 30(4):2177–2205, 2016. doi:10.1137/15M1034337.

22 Robert E Helbig, Patrick K Orr, and Robert R Roediger. Political redistricting by computer. *Communications of the ACM*, 15(8):735–741, 1972.

23 Gregory Herschlag, Han Sung Kang, Justin Luo, Christy Vaughn Graves, Sachet Bangia, Robert Ravier, and Jonathan C. Mattingly. Quantifying gerrymandering in North Carolina, 2018. arXiv:1801.03783.

24 Gregory Herschlag, Robert Ravier, and Jonathan C. Mattingly. Evaluating partisan gerrymandering in Wisconsin, 2017. arXiv:1709.01596.

25 S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006, 1965.

26 Takehiro Ito, Kazuya Goto, Xiao Zhou, and Takao Nishizeki. Partitioning a multi-weighted graph to connected subgraphs of almost uniform size. *IEICE Trans. Inf. Syst.*, 90-D(2):449–456, 2007. doi:10.1093/ietisy/e90-d.2.449.

27 Takehiro Ito, Xiao Zhou, and Takao Nishizeki. Partitioning a graph of bounded tree-width to connected subgraphs of almost uniform size. *J. Discrete Algorithms*, 4(1):142–154, 2006. doi:10.1016/j.jda.2005.01.005.

28 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *J. Comput. Syst. Sci.*, 79(1):39–49, 2013. doi:10.1016/j.jcss.2012.04.004.

29 Jun Kawahara, Takashi Horiyama, Keisuke Hotta, and Shin-ichi Minato. Generating all patterns of graph partitions within a disparity bound. In *International Workshop on Algorithms and Computation*, pages 119–131. Springer, 2017.

30 Philip N. Klein and Shay Mozes. Optimization Algorithms for Planar Graphs. `http://planarity.org/`. accessed June 2018.

31 Dániel Marx, Ario Salmasi, and Anastasios Sidiropoulos. Constant-factorapproximations for asymmetric TSP on nearly-embeddable graphs. In *Proceedings of the 19th Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 60 of *LIPIcs*, pages 16:1–16:54, 2016. `doi:10.4230/LIPIcs.APPROX-RANDOM.2016.16`.

32 Anuj Mehrotra, Ellis L. Johnson, and George L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114, 1998.

33 Lorenzo Najt, Daryl R. DeFord, and Justin Solomon. Complexity and geometry of sampling connected graph partitions. *CoRR*, abs/1908.08881, 2019. `arXiv:1908.08881`.

34 J. K. Park and C. A. Phillips. Finding minimum-quotient cuts in planar graphs. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 766–775, 1993. `doi:10.1145/167088.167284`.

35 W. Pegden. Pennsylvania's congressional districting is an outlier: Expert report, League of Women Voters vs. Pennsylvania General Assembly, 2017. URL: `https://www.brennancenter.org/sites/default/files/legal-work/LWV_v_PA_Expert_Report_WesleyPegden_11.17.17.pdf`.

36 Richard H Pildes, Tacy F Flint, and Sidley Austin. Brief of political geography scholars as amici curiae in support of appellees. URL: `https://www.brennancenter.org/sites/default/files/legal-work/Gill_AmicusBrief_%20Political%20Geography%20Scholars_InSupportofAppellees.pdf`.

37 Satish Rao. Finding near optimal separators in planar graphs. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 225–237, 1987. `doi:10.1109/SFCS.1987.26`.

38 Satish Rao. Faster algorithms for finding small edge cuts in planar graphs. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 229–240, 1992. `doi:10.1145/129712.129735`.

A Possibility in Algorithmic Fairness: Can Calibration and Equal Error Rates Be Reconciled?

Claire Lazar Reich ✉
MIT Statistics Center and Department of Economics, Cambridge, MA, USA

Suhas Vijaykumar ✉
MIT Statistics Center and Department of Economics, Cambridge, MA, USA

Abstract

Decision makers increasingly rely on algorithmic risk scores to determine access to binary treatments including bail, loans, and medical interventions. In these settings, we reconcile two fairness criteria that were previously shown to be in conflict: calibration and error rate equality. In particular, we derive necessary and sufficient conditions for the existence of calibrated scores that yield classifications achieving equal error rates at any given group-blind threshold. We then present an algorithm that searches for the most accurate score subject to both calibration and minimal error rate disparity. Applied to the COMPAS criminal risk assessment tool, we show that our method can eliminate error disparities while maintaining calibration. In a separate application to credit lending, we compare our procedure to the omission of sensitive features and show that it raises both profit and the probability that creditworthy individuals receive loans.

2012 ACM Subject Classification Mathematics of computing → Probability and statistics; Social and professional topics → Computing / technology policy; Computing methodologies → Supervised learning

Keywords and phrases fair prediction, impossibility results, screening decisions, classification, calibration, equalized odds, optimal transport, risk scores

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.4

Related Version *Full Version*: `https://arxiv.org/abs/2002.07676`

Funding *Claire Lazar Reich*: Supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374.

Acknowledgements Many thanks to Anna Mikusheva, Iván Werning, and David Autor for their valuable advice. We're also deeply grateful for the support of Ben Deaner, Lou Crandall, Pari Sastry, Tom Brennan, Jim Poterba, Rachael Meager, and Frank Schilbach with whom we have had energizing and productive conversations. Thank you to Deborah Plana, Pooya Molavi, Adam Fisch, and Yonadav Shavit for commenting on the manuscript at its advanced stages.

1 Introduction

Today's algorithms reach deep into decisions that guide our lives, from loan approvals to medical treatments to foster care placements. Making these high-impact decisions fairly is an effort undergoing public scrutiny. In one investigation, *ProPublica* showed that an algorithm operating in the U.S. criminal justice system, COMPAS, discriminated against black defendants by misclassifying them as high-risk at significantly higher rates than white defendants [2]. On the other hand, it was later revealed that the same algorithm did satisfy a different form of fairness: calibration of scores for both black and white defendants [10]. This meant that on average, a defendant's score reflected the same risk level regardless of race.

Researchers have sought to explain how a screening algorithm like COMPAS can satisfy one natural notion of fairness but not another, spurring a research agenda to characterize how definitions of algorithmic fairness relate to one another. Multiple studies in this literature

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 4; pp. 4:1–4:21
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

proved that algorithms face inevitable tradeoffs whenever they predict on groups that have different average outcomes [15, 5, 4, 7, 14]. These influential "impossibility results" have underscored the need for practitioners to target certain fairness criteria at the expense of others.

We show that it is in fact possible to reconcile the two notions of fairness that gained influence following the COMPAS investigation: calibration and equal error rates. In important previous work, these two criteria were proven to be mutually incompatible when both are applied to a *risk score* [15, 22] and when both are applied to a *classifier* [5]. Naturally these findings were interpreted as evidence that calibration and equal error rates are incompatible altogether [1]. It was therefore speculated that COMPAS's enforcement of score calibration made its error rate imbalances inevitable [5].

In contrast, we show that both calibration and equal error rates can be reconciled in COMPAS and in many other real-world settings where protected groups have different mean outcomes. We relax the mathematical tension between these two fairness criteria by separately enforcing *calibration on the score* and *equal error rates on the corresponding classifier*. In particular, we prove that it is possible to design calibrated scores that yield equal error rate classifications at group-blind cutoffs, and we provide a method to do so with maximal accuracy. Furthermore, we develop practical extensions of the method, such as showing how to enforce weaker notions of the equal error rate criterion (like the "equality of opportunity" criterion of Hardt et al. [12]) and how to accommodate multiple protected subgroups.

Our framework and method can be applied to two settings. In the first, we consider the problem of providing risk scores to a profit-maximizing third-party agent, such as a lender, who then uses them to assign binary treatments, such as loan approvals and denials. We illustrate how to construct calibrated scores that lead this profit-maximizer to make classifications satisfying equal error rates. In the second setting, we consider risk assessments like COMPAS that output both scores and classification recommendations, and show that the scores can be made to satisfy calibration while the classification recommendations can be made to satisfy equal error rates.

This paper supports growing evidence on the complementary relationship between data quality and fairness objectives [11, 8, 9, 6, 13, 14]. In particular, we show that access to sufficiently informative features is required to satisfy our fairness criteria, and that the feasible set of solutions grows with the informativeness of the data. In an empirical credit lending example, we compare our method to a commonly practiced strategy of data omission. It yields higher lender profit while also improving access to loans for creditworthy applicants in all groups.

The results proceed as follows. In Section 2, we prove that it is possible to construct calibrated scores that lead to equal error rate classifications and we precisely characterize when such scores exist. In Section 3, we propose an algorithm that produces the most accurate possible score satisfying the fairness criteria and minimizing the decision-maker's errors. We apply our method in Section 4 to two empirical settings. We first assess its performance in helping a lender screen loan applicants of various educational backgrounds. We also apply the method to the COMPAS criminal risk assessment tool, where we show that our procedure can eliminate error rate imbalances in risk classifications while preserving calibration of scores.

1.1 Related Work

Our paper belongs to a body of work that studies the mathematical relationships between various individual and group measures of fairness. Calibration and equal error rates have been formalized and extensively studied in prior work [22, 12, 15, 5]. In particular, Kleinberg et al. [15] and Pleiss et al. [22] show that these criteria are incompatible when applied to a risk score and Chouldechova [5] shows the corresponding result for binary classifiers. We consider a natural variation of the problem where we ask whether a calibrated score can, upon being supplied to a rational third-party, lead to equal-error predictions. Surprisingly, we find that the answer is yes.

Our work also contributes to a recent strand of the literature which studies how algorithmic prediction can interact with self-interested decision makers, bridging the classical problem of prediction with the traditionally economic problem of information design [20, 24]. From this perspective, we study the existence of scores that lead to desirable equilibria: those in which the final decision rule is group-blind due to calibration, and the resulting decisions satisfy equal error rates.

Finally, we believe it is important to emphasize that the two fairness criteria we study do not encompass all notions of fairness. Tradeoffs remain between these criteria and others. For example, enforcing equal error rates requires that the classifications' positive and negative predictive values will be unequal across groups, meaning that one groups' scores would carry greater signal to the decision-maker than the others' [5]. In addition, equal error rate classifications will generically require changes to the Bayes' optimal classifications, and enforcing calibration does not diminish this requirement [6].

Decisions for how to prioritize fairness conditions are likely to vary by application going forward. We hope that by clarifying the precise relationship between two influential criteria, we can facilitate these decisions, and that in settings where calibration and equal error rates are considered essential, our algorithm can help yield accurate predictions and fairer outcomes.

2 Theoretical Results

2.1 Formal Setting

Let us consider a triple (Y, X, A) on a common probability space $\mathbb{P}$, where $Y \in \{0, 1\}$ is an outcome variable, $X \in \mathbb{R}^d$ is a vector of features, and $A \in \{H, L\}$ is a protected attribute differentiating two groups with unequal base rates $\mu_A = \mathbb{E}[Y|A]$ of the outcome,

$$\mu_L < \mu_H. \tag{1}$$

Our goal is to estimate a score function $\hat{p} \equiv \hat{p}(X, A) \in [0, 1]$ that predicts Y with maximum accuracy subject to the constraints of calibration and equal error rates. Specifically, we hand $\hat{p}$ to a decision-maker tasked with selecting classifications $\hat{y} \in \{0, 1\}$ that minimize their loss function

$$\ell(\hat{y}, y) = \begin{cases} 0 & y = \hat{y} \\ 1 & y > \hat{y} \\ k & y < \hat{y}, \end{cases} \tag{2}$$

where $k > 0$ is the relative cost of false positive classifications. Note that any loss function that is minimized when $y = \hat{y}$ is equivalent to ℓ after an affine transformation.

Let us suppose the decision-maker might be able to observe group affiliation A in addition to $\hat{p}$. To ensure that classifications are based only on $\hat{p}$ and not on A, we constrain $\hat{p}$ to satisfy *calibration within groups*,

$$\mathbb{E}[Y|A,\hat{p}] = \mathbb{E}[Y|\hat{p}] = \hat{p}. \tag{3}$$

If (3) holds, the decision-maker's expected loss given $\hat{p}$ and A becomes

$$\mathbb{E}[\ell(Y,\hat{y})|\hat{p},A] = \hat{p}(1-\hat{y}) + k(1-\hat{p})\hat{y}. \tag{4}$$

This expected loss is minimized with a cutoff decision rule that is independent of group affiliation A,

$$\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}, \tag{5}$$

where the cutoff $\bar{p} = {}^{k}\!/\!_{(k+1)}$ is fixed by the decision-maker's loss function.

Our second condition constrains $\hat{y}$ to satisfy *equal error rates*, ensuring that the classification only depends on the group through the target variable. Following the decision rule (5), we may write this as

$$(\mathbb{1}\{\hat{p} \geq \bar{p}\} \perp\!\!\!\perp A) \mid Y. \tag{6}$$

Our calibration and equal error rate conditions are summarized by (3) and (6), respectively.

2.2 Relation to Impossibility Results

We first introduce a general impossibility result, relate it to previous work, and show where our assumptions diverge to make our proposed criteria satisfiable. The following theorem proves that a *single* algorithmic output Z cannot generally satisfy notions of both calibration and equal error rates.

▶ **Theorem 1.** *Let Y, A, and Z be random variables satisfying the following three conditions.*

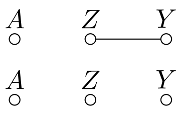
(i) $(Y \perp\!\!\!\perp A) \mid Z$,

(ii) $(Z \perp\!\!\!\perp A) \mid Y$,

(iii) $\mathbb{P}(A = H|Z)$, $\mathbb{P}(Y = 1|A, Z) \in (0, 1)$.

Then A and (Z, Y) must be independent.

Proof. Suppose that (Y, A, Z) satisfy (i) (ii) and (iii). Assumption (iii) implies that the law of (A, Y, Z) is strictly positive. By the Hammersley-Clifford theorem (see e.g. [18]), the conditional independence relations are summarized by a graph on $\{Y, A, Z\}$ where every path from Y to A travels through Z, and every path from A to Z travels through Y. There are only two graphs with this property:



In neither of these graphs does there exist a path from A to (Y, Z), so we conclude that A and (Y, Z) must be independent for (i) (ii) and (iii) to simultaneously hold. ◀

Note that when A denotes group affiliation and Y denotes outcomes, (i) is a form of calibration and (ii) is a form of the equal error rate condition. Assumption (iii) is a strong form of predictive uncertainty that is generalized in the appendix. Thus the theorem shows that when there is predictive uncertainty and Y depends on A (i.e. when the base rates are unequal), it is impossible for a single Z to satisfy both calibration and equal error rates. For example, letting Z be a classifier recovers the result of Chouldechova that (i) equal positive and negative predictive values are unachievable alongside (ii) equal error rates [5]. Meanwhile, letting Z be a risk score shows that (i) calibration is unachievable alongside (ii) a condition that implies balance in the positive and negative class, similar to the result of Kleinberg et al. [15].

Our own setting bypasses the mathematical impossibility described in Theorem 1 by imposing constraints on *two* separate algorithmic outputs rather than one. We require (i) calibration from the scores $\hat{p}$ and (ii) equal error rates from the resulting classifications $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$.

2.3 Necessary and Sufficient Conditions

In this section we characterize exactly when there exists a calibrated $\hat{p}$ that leads to equal error rate classifications $\hat{y}$ at the cutoff $\bar{p}$. Our conditions can be easily checked in a given setting, and they are shown to depend on the informativeness of the features X.

The graphical framework in this section builds on methods developed by Hardt et al. [12]. All the necessary and sufficient conditions will be illustrated in $\mathbb{R}^2$, with true positive rates on the vertical axis and false positive rates on the horizontal. The *feasible region* will be the set in $\mathbb{R}^2$ corresponding to error rates achievable by an equal error rate classifier $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$ where $\hat{p}$ is calibrated.

We first study the entire set corresponding to equal error rate classifiers, without regard to calibration or the decision-maker's cutoff $\bar{p}$. Then we study the entire set corresponding to classifiers that can be based on the cutoff $\bar{p}$ applied to calibrated scores, without regard to the equal error rate condition. Finally, we prove that the intersection of these two sets determines feasibility of enforcing both conditions, and we characterize when the intersection is nonempty.

2.3.1 Classifiers Satisfying Equal Error Rates

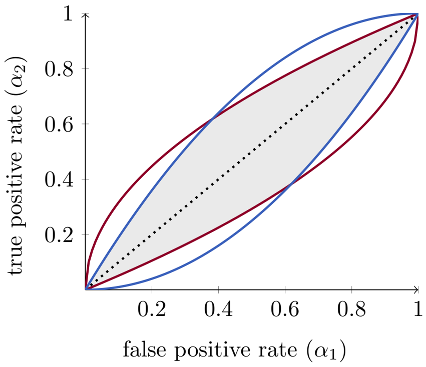
We wish to identify the entire set of error rates in $\mathbb{R}^2$ achievable by classifiers with equal error rates. Hardt et al. [12] succeeded in doing so, and we review and adapt their results in this subsection. To lay the groundwork for the geometric reasoning to follow, we first denote the group A false positive rate and true positive rate associated with a given classifier $\hat{y}$ as a point in $\mathbb{R}^2$,

$$\alpha(\hat{y}, A) = \Big(\mathbb{P}(\hat{y} = 1 | Y = 0, A), \mathbb{P}(\hat{y} = 1 | Y = 1, A)\Big).$$

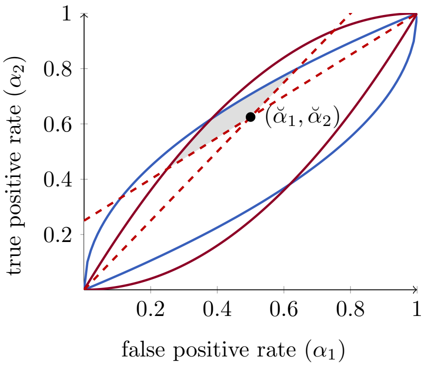
We may now define the set of achievable error rates in $\mathbb{R}^2$. Let $\mathcal{H}$ be the set of all possibly random classifiers $h(X, A)$. The set of achievable error rates for group A is

$$S(A) = \{\alpha(\hat{y}, A) \,|\, \hat{y} = h(X, A), h \in \mathcal{H}\} \subseteq \mathbb{R}^2, \tag{7}$$

and the set of achievable rates for all classifiers satisfying equal error rates is given by $S(L) \cap S(H)$. To better understand this intersection, we characterize $S(A)$ in terms of Receiver Operator Characteristic (ROC) curves following Hardt et al. [12]. By definition, an ROC curve of a given score p traces the true and false positive rates associated with each possible cutoff rule $\mathbb{1}\{p \geq c\}$ for $c \in [0, 1]$. Therefore it contains all points $\alpha(\mathbb{1}\{p \geq c\}, A)$. With these tools in hand, we are ready to characterize the feasible set of rates $S(A)$ for group A.



🟨 **Figure 1** Achievable equal error rates (shaded). Two pairs of ROC curves form the boundaries of $S(L)$ and $S(H)$. Points in the intersection $S(L) \cap S(H)$ correspond to equal error rate classifiers.



🟨 **Figure 2** Achievable equal error rates from calibrated score at cutoff $\bar{p}$ (shaded). The restrictions (11) correspond to half-spaces above the red dashed lines.

▶ **Proposition 2.** *Let $p^* = p^*(X, A)$ be the Bayes optimal score satisfying $p^* = \mathbb{E}[Y|X, A]$, i.e., the best score given our data. Then the set of achievable rates $S(A)$ is exactly the convex hull of the union of the group-A ROC curve of the best score p^* and the group-A ROC curve of the worst score $1 - p^*$, i.e. the convex hull of*

$$\Big\{\alpha(\mathbb{1}\{p^* \geq c\}, A) \,\Big|\, 0 \leq c \leq 1\Big\} \\ \cup \Big\{(1,1) - \alpha(\mathbb{1}\{p^* \geq c\}, A) \,\Big|\, 0 \leq c \leq 1\Big\}.$$

Figure 1 illustrates typical examples of $S(L)$, $S(H)$, and the intersection $S(L) \cap S(H)$ which represents the rates achievable by equal error rate classifiers.

2.3.2 Classifiers Compatible with Calibration

We now put aside the equal error rate constraint and concentrate on identifying the entire set of classifiers that are implementable with the cutoff $\bar{p}$ applied to some calibrated scores $\hat{p}$. The set is characterized by the following proposition.

▶ **Proposition 3.** *A classifier $\hat{y}$ can be written as $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$ for some calibrated $\hat{p}$ if and only if its group-specific positive predictive values exceed $\bar{p}$, and its group-specific negative predictive values exceed $1 - \bar{p}$. In particular, for $A \in \{L, H\}$,*

$$\mathbb{P}(Y = 1|\hat{y} = 1, A) \geq \bar{p}, \quad \mathbb{P}(Y = 0|\hat{y} = 0, A) > 1 - \bar{p}. \tag{8}$$

Proof. Suppose that $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$ where $\hat{p}$ is calibrated. Then $\hat{y}$ must satisfy the inequalities

$$\begin{aligned} \mathbb{P}(Y = 1|\hat{y} = 1, A) &= \mathbb{E}[Y|\hat{p} \geq \bar{p}, A] \\ &= \mathbb{E}[\hat{p}|\hat{p} \geq \bar{p}, A] \geq \bar{p}, && (9) \\ \mathbb{P}(Y = 1|\hat{y} = 0, A) &= \mathbb{E}[\hat{p}|\hat{p} < \bar{p}, A] < \bar{p}. && (10) \end{aligned}$$

Therefore, if $\hat{y}$ is based on a calibrated score $\hat{p}$ at cutoff $\bar{p}$, then it is necessary for the group-specific positive and negative predictive values to exceed $\bar{p}$ and $(1 - \bar{p})$, respectively.

Conversely, given *any* classifier $\hat{y}$ that satisfies the inequalities (9) and (10), we can always put

$$\hat{p}(\hat{y}, A) = \mathbb{P}(Y = 1|\hat{y}, A)$$

to obtain a calibrated score that takes just two possible values per group with the cutoff $\bar{p}$ guaranteed to be between them. This choice of $\hat{p}$ thus satisfies $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$ by construction. ◀

As we will see in the following subsection, this result lays the foundation for the necessary and sufficient conditions for the satisfiability of our fairness criteria.

2.3.3 The Feasibility Region

Proposition 3 demonstrates that the following are equivalent:

(i) There exists a calibrated score $\hat{p}$ such that $\hat{y} = \mathbb{1}\{\hat{p} \geq \bar{p}\}$ satisfies equal error rates.

(ii) There exists a classifier $\hat{y}$ satisfying equal error rates and (8).

In practice, we propose checking (ii) to identify whether (i) holds. To do so, we use Bayes' rule to write (8) as group-specific restrictions on true and false positive rates so that we can consider them in the same space as the equal error rate constraints given by Hardt et al. [12]. The following theorem and the accompanying Figure 2 indicate that each restriction (8) corresponds to a half-space in $\mathbb{R}^2$, and that the feasibile region corresponds to the intersection of those half-spaces with each other and with the equal error rates region $S(L) \cap S(H)$.

▶ **Theorem 4.** *Let $\beta_A = \mu_A/(1 - \mu_A)$ denote the group-specific odds ratios, with $\beta_L < \beta_H$. Then our fairness criteria are simultaneously satisfiable at cutoff $\bar{p}$ if and only if there exists $(\alpha_1, \alpha_1) \in S(L) \cap S(H)$ satisfying the two inequalities*

$$\frac{\alpha_2}{\alpha_1} \geq \frac{\bar{p}}{\beta_L(1 - \bar{p})}, \quad \frac{(1 - \alpha_1)}{(1 - \alpha_2)} > \frac{\beta_H(1 - \bar{p})}{\bar{p}}. \tag{11}$$

We next provide easily checkable necessary and sufficient conditions for when the feasible region is nonempty.

▶ **Corollary 5.** *Let $(\breve{\alpha}_1, \breve{\alpha}_2)$ denote the point at which the inequalities* (11) *hold with equality. Our fairness criteria are simultaneously satisfiable at cutoff $\bar{p}$ if and only if any of the following holds: $\breve{\alpha}_1 \leq 0$, $\breve{\alpha}_1 \geq 1$, or both groups' ROC curves corresponding to p^* lie above $(\breve{\alpha}_1, \breve{\alpha}_2)$. Note that $(\breve{\alpha}_1, \breve{\alpha}_2)$ are fixed by the group base rates and decision-maker's cutoff $\bar{p}$,*

$$\breve{\alpha}_1 = \frac{\beta_L}{(\beta_H - \beta_L)} \left(\frac{\beta_H - (1 + \beta_H)\,\bar{p}}{\bar{p}} \right), \qquad \breve{\alpha}_2 = \frac{1}{(\beta_H - \beta_L)} \left(\frac{\beta_H\,(1 - \bar{p}) - \bar{p}}{1 - \bar{p}} \right). \tag{12}$$

We note that the feasible region depends on the decision-maker's cutoff $\bar{p}$, which in turn depends on their relative valuation of false positive and false negative classifications, k. In particular, when k is either very large or close to 0, the set of feasible error rates shrinks to include only those corresponding to no positive classifications or no negative classifications.

Data quality also contributes to the feasibility of enforcing both fairness criteria, as illustrated by Theorem 4 and Figure 2. Note that the intersection of the half-spaces defined in (11) are fixed by given parameters: β_L, β_H, and $\bar{p}$. Beyond these, what determines the size of the feasible region is the height of the ROC curves.

Higher ROC curves correspond to more accurate predictions, which can be achieved by including more informative features X. This expands the region $S(H) \cap S(L)$ and thus always weakens the constraints dictating whether equal error rates and calibration are compatible in a given setting. Therefore, increasing the quality of data that an algorithm can access promotes our notions of fairness, whereas removing data compromises them.

3 A Loss-Minimizing Algorithm

After checking that our fairness criteria are feasible in a given setting, a natural next step is to search for the constrained optimal solution, i.e. to identify the most accurate score $\hat{p}$ that minimizes the decision-maker's loss subject to our fairness constraints. Our strategy is to first estimate the most accurate score $p^* = \mathbb{E}[Y|X, A]$ without regard to fairness, and then to transform the estimate in two separate stages. First, we identify the error rates that minimize loss subject to the fairness conditions (Section 3.1). Second, we identify the MSE-minimizing calibrated scores $\hat{p}$ that gives rise to those error rates at the decision-maker's cutoff $\bar{p}$ (Section 3.2). Lastly, we lay out extensions of the algorithm that can accommodate practical use cases (Section 3.3).

3.1 Stage 1: Error Rate Optimization

The first stage of the algorithm identifies feasible error rates that minimize the decision maker's loss.

Let R denote the set of points (α_1, α_2) in the feasible region, i.e. the pairs of error rates in $S(H) \cap S(L)$ that satisfy (11). Note that R is necessarily convex, as it is the intersection of four convex regions: $S(H)$, $S(L)$, and the half-spaces defined in (11). Moreover, according to the decision-maker's loss function, a classifier corresponding to error rates (α_1, α_2) obtains expected loss

$$\ell(\alpha_1, \alpha_2) \equiv k\alpha_1(1 - \mathbb{E}[Y]) + (1 - \alpha_2)\mathbb{E}[Y]. \tag{13}$$

Thus, straightforward convex optimization will identify the error rates that minimize the linear function ℓ over $(\alpha_1, \alpha_2) \in R$. The optimal error rates identified, $z^* = (\alpha_1^*, \alpha_2^*)$, will be on the upper-left boundary of the feasible region in Figure 2, with the precise point determined by the decision-maker's relative preference k over false positive and false negative classifications.

Algorithm Stage 1 Find loss-minimizing feasible error rates.

```
Input: Raw scores $\{p_i^*\}$, labels $\{Y_i\}$, group identities $\{A_i\}$, base rates $\mu_A$, cutoff $\bar{p}$, loss parameter $k$.
Step 1: Define convex feasible region $R$ by taking intersection of rates $(\alpha_1, \alpha_2)$ in $S(L) \cap S(H)$ that satisfy (11). To compute $S(L) \cap S(H)$, use $\{p_i^*\}$ to determine each group's ROC curves.
if $R$ is empty then
  Output: No feasible solution.
end if
Step 2: Minimize loss function (13) over $(\alpha_1, \alpha_2) \in R$.
Output: Optimal target rates $(\alpha_1^*, \alpha_2^*)$ from Step 2.
```

▶ Remark 6. The sets $S(L)$ and $S(H)$ correspond to the Bayes optimal score $p^* = \mathbb{E}[Y|X, A]$, which needs to be estimated in practice. Given an estimated score p, we propose using a holdout sample to first calibrate p and then perform our algorithm. The resulting scores will satisfy the fairness criteria approximately by a law-of-large-numbers argument, where the fidelity is determined solely by the size of the holdout sample (see e.g. [27]).

3.2 Stage 2: Risk Score Optimization

Once a feasible set of error rates is chosen, the decision-maker's expected loss is determined. However, multiple choices of calibrated scores may achieve those target rates at the cutoff $\bar{p}$, and we expect that in practice, decision-makers would prefer more accurate scores. This section thus describes a method to recover the MSE-minimizing score $\hat{p}$ that implements the target rates z^* by solving a constrained optimal transport problem [21].

We base the method on the finding that the best $\hat{p}$ satisfying the fairness criteria is recoverable through post-processing the Bayes optimal score $p^* = \mathbb{E}[Y|X, A]$. We include a proof for this in the appendix, following a similar argument of Hardt et al. [12]. In the appendix we also discuss how our procedure can be thought of as finding the smallest *mean-preserving contraction* of p^* that yields the targeted error rates. Readers will note that the post-processing procedure requires some randomization of input scores. We explore the effects of the randomization empirically in our online appendix [23], and meanwhile highlight that our algorithm's accuracy objective limits the extent to which scores p^* change.

Our method defines one linear program per group A and seeks the most accurate $\hat{p}_A$ that yields error rates at the cutoff $\bar{p}$ given by

$$\alpha(\mathbb{1}\{\hat{p}_A \geq \bar{p}\}, A) = z^* = (\alpha_1^*, \alpha_2^*).$$

For the remainder of the section, we simplify notation by suppressing A subscripts and note that the procedure is performed once for each group $A \in \{H, L\}$.

Our approach will involve a transformation kernel, or *transport map*, that maps the distribution of the most accurate estimate of p^* to the distribution of our post-processed $\hat{p}$. We assume for simplicity that the p^* estimate has already been calibrated, and that it is discrete (which we justify in the appendix). In particular, p^* takes N ordered values $p = (p_1, p_2, \ldots, p_N)$, each with probability mass given by $s = (s_1, s_2, \ldots, s_N)$ where $\sum_i s_i = 1$. Furthermore, we will denote the post-processed $\hat{p}$ as taking those same discrete values p but with different probability masses that we seek to optimize, $f = (f_1, f_2, \ldots, f_N)$.

We call T the matrix that maps probability masses from the discrete distribution of p^* to that of $\hat{p}$. In particular, with probability T_{ij}, the kernel will map an individual with score p_i to the output score p_j. Therefore, the probability distribution of $\hat{p}$ will be determined by

$$T^\top s = f. \tag{14}$$

In order to produce probability distributions, T must be right-stochastic: elements must take values between 0 and 1, and each row should sum to 1.

$$0 \leq T_{ij} \leq 1 \text{ and } \sum_{k=1}^{N} T_{ik} = 1 \quad \forall\, i, j \in \{1, \ldots N\}. \tag{15}$$

According to our fairness criteria, we further constrain T. To ensure that $\hat{p}$ will be calibrated, we need the outcome of individuals assigned score f_i to satisfy $Y = 1$ with probability p_i. Given our assumption that p^* has itself been calibrated, this reduces to

$$\sum_{i=1}^{N} T_{ij} p_i s_i = p_j f_j \quad \forall\, j \in \{1, \ldots, N\}. \tag{16}$$

The targeted false- and true-positive rates (α_1^*, α_2^*) derived in Section 3.1 similarly require:

$$\sum_{j=1}^{N}\sum_{i=1}^{N} T_{ij} p_i s_i \left(\mathbb{1}\{p_j \geq \bar{p}\} - \alpha_2^*\right) = 0,$$
$$\sum_{j=1}^{N}\sum_{i=1}^{N} T_{ij} (1 - p_i) s_i \left(\mathbb{1}\{p_j \geq \bar{p}\} - \alpha_1^*\right) = 0. \tag{17}$$

Finally, we formulate an objective. Note that the mean-squared error of $\hat{p}$ satisfies the bias-variance decomposition

$$\mathbb{E}[(\hat{p} - Y)^2] = \mathbb{E}[(\hat{p} - \mathbb{E}[Y|X, A])^2] + \mathbb{E}[(Y - \mathbb{E}[Y|X, A])^2],$$

and thus the $\hat{p}$ that minimizes the left hand side is obtained by minimizing the first term on the right hand side. In particular, if the input score p^* is $\mathbb{E}[Y|X, A]$, then the post-processed score that minimizes mean-squared error will also minimize

$$\mathbb{E}[(\hat{p} - p^*)^2] = \sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij} (p_i - p_j)^2 s_i. \tag{18}$$

Furthermore, even if p^* is not exactly equal to $\mathbb{E}[Y|X, A]$, the triangle inequality in $L^2(\mathbb{P})$ implies

$$\mathbb{E}[(\hat{p} - Y)^2]^{\frac{1}{2}} \leq \mathbb{E}[(p^* - Y)^2]^{\frac{1}{2}} + \mathbb{E}[(\hat{p} - p^*)^2]^{\frac{1}{2}}.$$

Thus, by minimizing the objective (18) we can effectively control the additional error due to post-processing. Combining this with the above constraints yields a straightforward linear program.

Algorithm Stage 2 For each group, find calibrated scores achieving target rates.

Input: Raw scores $\{p_i^*\}$, number of bins N, target error rates from stage 1 of algorithm (α_1^*, α_2^*), cutoff $\bar{p}$.

Step 1: Produce discrete score approximation of p^*: label N ordered values $(p_1, p_2, \ldots, p_N)$ with masses $(s_1, s_2, \ldots, s_N)$.

Step 2: Find score transformation kernel T that minimizes (18) subject to the constraints (14), (15), (16) and (17).

Step 3: Map each individual with given raw score to a new post-processed score, based on probabilities given by kernel T.

Output: Scores $\hat{p}$ from Step 3. By design these satisfy calibration and yield error rates (α_1^*, α_2^*) at cutoff $\bar{p}$.

3.3 Available Extensions

Our procedure can be modified to handle additional use cases. We can flexibly trade off the fairness and accuracy objectives, minimize error disparities rather than eliminate them when the feasible region is empty, accommodate a setting where the decision-maker's cutoff $\bar{p}$ is estimated with error, and apply the procedure to more than two groups.

3.3.1 Relaxing the fairness criteria

An alternative formulation of our algorithm can accommodate multiple cases encountered in practice. By modifying Stage 1 to include a weighted error-rate penalty, users can flexibly trade off the fairness and accuracy objectives, minimize error disparities rather than eliminate them when the feasible region R is empty, and enforce just one error constraint as in the "equality of opportunity" criterion of Hardt et al. [12]. In general, the more flexible procedure will output group-specific optimal error rates: z_L^* and z_H^*. These group-specific targets are then inputted into Stage 2 which is otherwise unchanged.

To modify Stage 1, first we define a broader domain for the algorithm to search over in place of R. It contains all the error rates implementable by a calibrated score at the decision-maker's cutoff, according to the inequalities (11), without regard to equal error rates. The domain is $R(H) \times R(L)$ where

$$R(A) = \left\{ (\alpha_1, \alpha_2) \in S(A) \,\middle|\, \frac{1-\alpha_2}{1-\alpha_1} < \frac{\bar{p}/\beta_A}{(1-\bar{p})} \leq \frac{\alpha_2}{\alpha_1} \right\}. \tag{19}$$

(Note that this is guaranteed to be nonempty, as it contains the error rates of the classifier $\mathbb{1}\{p^* \geq \bar{p}\}$.) We also replace the loss function (13) with a generalized version that includes both the decision-maker's expected loss from the error rates as well as the groups' rate disparities. The new loss function is

$$\gamma \ell(z_L) + (1-\gamma)\ell(z_H) + (z_L - z_H)^\top \Lambda (z_L - z_H) \tag{20}$$

where $\ell(z_A)$ is the decision-maker's expected loss $k\alpha_{1A}(1 - \mathbb{E}[Y|A]) + (1-\alpha_{2A})\mathbb{E}[Y|A]$ and γ is the fraction of individuals in group L. Meanwhile, Λ is a positive semidefinite matrix that provides the flexibility of varying the enforcement of minimal error rate differences. For example, taking $\Lambda = \lambda I$ for arbitrarily large λ recovers the equal error rate solution when the feasible region R is nonempty, and otherwise outputs the solution that minimizes error rate disparities. Meanwhile a small choice of λ places relatively more weight on accuracy.

Alternatively, Λ could be chosen so that differences in the true and false positive rates are weighted differently. For example, we can achieve equal true positive rates and allow false positive rates to vary [12] by letting $\Lambda(2,2)$ be large and assigning 0 to all other entries in Λ.

As a result of the flexible procedure, group-specific error rates z_L^* and z_H^* are identified to minimize the generalized loss function (20). The second stage of the algorithm can then be applied to identify a calibrated score that yields those target rates.

3.3.2 Accommodating an interval of possible k or $\bar{p}$

In settings where the exact $\bar{p}$ is unknown or not fixed, users can adapt our algorithm to function for any cutoff in an interval $(\bar{p}-\epsilon, \bar{p}+\epsilon)$. It can be tailored to produce scores $\hat{p}$ that are either below $\bar{p}-\epsilon$ or above $\bar{p}+\epsilon$, so that any cutoff applied within the interval would execute the same classifications.

In particular, we propose a couple modifications to generalize our algorithm to this setting. We wish for anyone receiving scores above $\bar{p}+\epsilon$ to be classified as $\hat{y}=1$ and anyone receiving scores below $\bar{p}-\epsilon$ as $\hat{y}=0$. Following the reasoning in Proposition 3, for such a score to be calibrated, the associated PPV should exceed $\bar{p}+\epsilon$ and the NPV should exceed $1-(\bar{p}-\epsilon)$. Therefore, the feasible region previously defined in Theorem 4 by (11) is now defined by the points $(\alpha_1, \alpha_1) \in S(L) \cap S(H)$ that satisfy

$$\frac{\alpha_2}{\alpha_1} \geq \frac{\bar{p}+\epsilon}{\beta_L(1-(\bar{p}+\epsilon))}, \quad \frac{(1-\alpha_1)}{(1-\alpha_2)} > \frac{\beta_H(1-(\bar{p}-\epsilon))}{\bar{p}-\epsilon}. \tag{21}$$

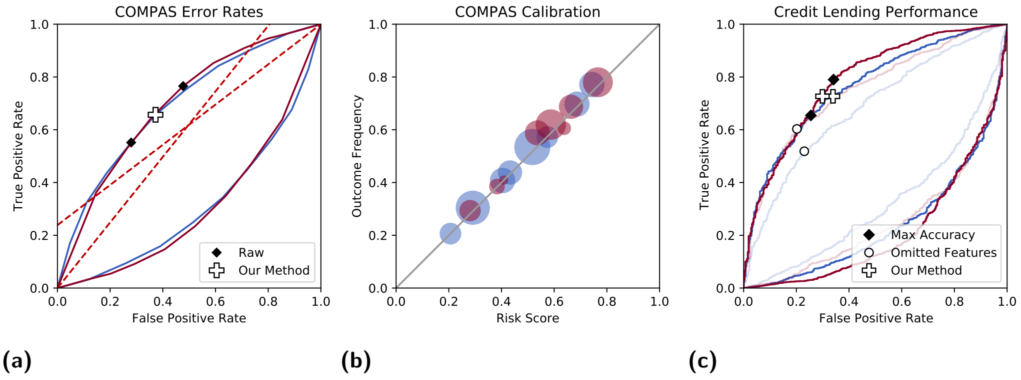


Figure 3 Evaluating algorithm performance. In each figure, maroon represents the high-mean group while blue represents the low-mean group. Panels (a) and (b) correspond to the criminal justice application, showing respectively that we can eliminate error rate disparities and maintain score calibration in COMPAS. Note that we define a true positive classification as correctly identifying someone who would not reoffend. Panel (c) covers the credit lending application, illustrating the empirical ROC curves from the rich feature set (opaque) and the limited feature set (translucent). Compared to a data omission strategy, our method raises the probability that creditworthy individuals from all education groups access loans.

This feasible region is used in Stage 1. In Stage 2, we add another constraint to specify that no post-processed scores be assigned values inside the interval of possible cutoffs:

$$T_{ik} = 0 \quad \forall k \text{ such that } p_k \in (\bar{p} - \epsilon, \bar{p} + \epsilon). \tag{22}$$

The rest of the procedure remains unchanged. The cost of the added flexibility is a tighter feasible region and higher MSE of the final score.

3.3.3 Satisfying the criteria for more than two protected groups

The algorithm can be modified to satisfy the fairness criteria for multiple groups, across multiple identifiers. First define each group as a unique combination of protected features. Then, the feasible set of error rates is given by the intersection of each $S(A)$ with the points satisfying the inequalities (11) where H is the highest-mean group and L is the lowest-mean group. Stage 1 of the algorithm proceeds to find the optimal set of error rates in that feasible region. Stage 2 proceeds as usual, implementing a separate program for each group.

4 Empirical Results

Let us take our procedure to data. In the first application, we post-process real COMPAS scores to demonstrate that risk assessments can be designed to output both calibrated risk scores as well as binary risk summaries satisfying equal error rates. Afterwards, we design a risk score to aid a lender's classification task to authorize loans, showing that it outperforms a common alternative strategy based on the omission of sensitive features. For interested readers, extensive detail about each application is presented in our online appendix [23].

4.1 Predicting criminal recidivism

Our procedure can design risk assessments that output both calibrated scores as well as binary "high" or "low" risk summaries satisfying equal error rates. We illustrate this in our first application, where we modify real criminal justice risk scores from COMPAS. As noted earlier, a *ProPublica* investigation showed that current COMPAS scores yield error imbalances across race, although they satisfy predictive parity overall [2, 1].

To check whether we can correct COMPAS error imbalances without sacrificing score calibration, we applied our post-processing technique to Broward County risk scores made public by *ProPublica* [16]. We define the outcome of interest as recidivism within two years, and we convert existing COMPAS scores that range from $[1, 10]$ to probabilities in $[0, 1]$. We define the classification cutoff as the minimum score of defendants classified as "high risk" in COMPAS, according to *ProPublica*'s influential analysis [17]. This corresponds to a cutoff of $\bar{p} = 0.54$ and loss parameter $k = 1.17$.

We compute the feasible region of achievable error rates according to Stage 1 of our algorithm and identify the loss-minimizing pair, as depicted in Figure 3a. Then, we use Stage 2 to post-process the COMPAS scores to achieve new calibrated scores yielding that optimal pair of error rates. The calibration of our scores is depicted in Figure 3b, where we group together by race defendants with the same post-processed scores and show that their corresponding recidivism outcomes lie on the main diagonal. Overall, our procedure eliminates the reported error disparities across racial groups (Figure 3a) while also preserving calibration (Figure 3b).

4.2 Predicting loan repayment

We next present an example of designing a risk score to inform a credit lender's approvals of loan applicants. Our goal is to deliver to the lender calibrated scores for applications from two groups—one highly educated (H) and another less educated (L)—while ensuring that they yield classifications with equal group TPRs at the lender cutoff. That way, we know that qualified applicants will have the same probability of receiving a loan regardless of their education level. We suppose the lender in question views defaulting as highly costly and only authorizes loans to individuals with calibrated scores greater than $\approx .9$, corresponding to loss parameter $k = 10$.

We simulate this scenario by applying our algorithm to the Survey of Income and Program Participation (SIPP), a nationally-representative survey of the civilian population spanning multiple years [25]. We select as our outcome the ability to pay rent, mortgage, and utilities in 2016, and predict that outcome using survey responses from two years prior. We label individuals with at most a secondary school education as L and those with higher education as H.

The full dataset contains over 1,800 features spanning detailed financial variables (including work history, assets, and debts), as well as sensitive features (including demographic information). We apply our algorithm to the full feature set and derive calibrated scores that yield equal TPRs at the lender's cutoff, using our algorithm extension that allows FPRs to vary. Then, we compare its performance to two accuracy-maximizing procedures: one based on the full feature set, and another commonly-practiced approach based on the omission of sensitive features. The results are summarized numerically in Table 1 and graphically in Figure 3c. Compared to prediction on all features and no post-processing, our algorithm raises the TPR of L and lowers that of H, while raising lender loss. Meanwhile, compared to the commonly used data omission strategy, our algorithm raises the probabilities that creditworthy applicants from *both* education groups are granted loans, and lowers loss for the lender.

Table 1 Application to credit lending. Row [1] is based on raw scores. Row [2] summarizes the classifier that minimizes lender loss subject to equal true positive rates, given by the equal opportunity algorithm in Hardt et al. (2016). Row [3] summarizes our algorithm, which produces a calibrated score corresponding to equal true positive rate classifications; since it retrieves the same error rates as row [2], we see there is no added loss from enforcing score calibration. Row [4] summarizes the scores from the alternative procedure that omits sensitive features, displaying greater loss for the lender, lower true positive rates for both groups, and substantial error disparities across groups.

	Algorithmic Target	Lender Loss	TPR (H/L)	FPR (H/L)	Score MSE
	Trained on all features				
[1]	Accuracy Maximizing	.517	(.795/.661)	(.341/.255)	.072
[2]	Eq. TPR Only	.532	(.727/.727)	(.299/.339)	N/A
[3]	*Eq. TPR + Calibration*	.532	(.727/.727)	(.299/.339)	.073
	Trained on limited features				
[4]	Accuracy Maximizing	.591	(.603/.518)	(.202/.230)	.077

5 Conclusion

Decision-makers stand to benefit from algorithmic predictions. This paper studies fair prediction in the widespread setting in which a risk score is constructed to aid their classification tasks. We prove that it is possible to construct calibrated scores that lead to equal error rate classifications at group-blind cutoffs. We characterize exactly when it is possible and propose an algorithm that produces the most accurate score satisfying the fairness criteria and minimizing the decision-maker's errors. Compared to a commonly practiced strategy of omitting sensitive data, we show that our algorithm can produce scores that enhance both efficiency and equity.

References

1 Julia Angwin and Jeff Larson. Bias in Criminal Risk Scores Is Mathematically Inevitable, Researchers Say. *ProPublica*, 2016.

2 Julia Angwin and Jeff Larson. Machine Bias. *ProPublica*, 2016.

3 Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 104(3):671–732, 2016.

4 Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.

5 Alexandra Chouldechova. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2):153–163, 2017. doi:10.1089/big.2016.0047.

6 Sam Corbett-Davies and Sharad Goel. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *arXiv:1808.00023 [cs]*, 2018. arXiv:1808.00023.

7 Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 797–806, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3097983.3098095.

8 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.

9 Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 119–133, New York, NY, USA, February 2018. PMLR.

10 Avi Feller, Emma Pierson, Sam Corbett-Davies, and Sharad Goel. A computer program used for bail and sentencing decisions was labeled biased against blacks. it's actually not that clear. *The Washington Post*, 17, 2016.

11 Sumegha Garg, Michael P. Kim, and Omer Reingold. Tracking and improving information in the service of fairness. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 809–824, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3328526.3329624.

12 Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. Curran Associates, Inc., 2016.

13 Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic Fairness. *AEA Papers and Proceedings*, 108:22–27, 2018. doi:10.1257/pandp.20181018.

14 Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: Implications for fairness, stereotypes, and interpretability. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pages 807–808, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3328526.3329621.

15 Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 43:1–43:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.43.

16 Jeff Larson. Data and analysis for "Machine bias". *GitHub*, June 2017.

17 Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How We Analyzed the COMPAS Recidivism Algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

18 Marc Mezard and Andrea Montanari. *Information, Physics, and Computation.* Oxford University Press, Inc., USA, 2009.

19 Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8(1):141–163, 2021. doi:10.1146/annurev-statistics-042720-125902.

20 Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7599–7609. PMLR, 13–18 July 2020. URL: http://proceedings.mlr.press/v119/perdomo20a.html.

21 Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

22 Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On Fairness and Calibration. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5680–5689. Curran Associates, Inc., 2017.

23 Claire Lazar Reich and Suhas Vijaykumar. A possibility in algorithmic fairness: Calibrated scores for fair classifications, 2020. arXiv:2002.07676.

24 Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 13–18 July 2020. URL: http://proceedings.mlr.press/v119/shavit20a.html.

25 U.S. Census Bureau. Survey of income and program participation, 2014. URL: https://www.census.gov/programs-surveys/sipp/data/datasets.html.

26 Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

27 Marten Wegkamp. Model selection in nonparametric regression. *The Annals of Statistics*, 31(1):252–273, 2003. doi:10.1214/aos/1046294464.

28 Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1171–1180, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. doi:10.1145/3038912.3052660.

A Appendix

A.1 Addendum to Theorem 1

Addendum. We relax condition (iii) of Theorem 1 and replace it with the weaker condition that $\mathrm{Var}(Y|Z) > \epsilon$ almost surely. This will correspond to the assumption that Y cannot be perfectly predicted from any realization of Z.

We will make use of the criterion that Borel random variables R and R' are independent conditional on Σ iff for all bounded, continuous f and g we have

$$\mathbb{E}[f(R)g(R')|\Sigma] = \mathbb{E}[f(R)|\Sigma]\mathbb{E}[g(R')|\Sigma].$$

Now suppose that (Z, A, Y) are known to satisfy Theorem 1 conditions (i) and (ii), and that $\mathrm{Var}(Y|Z) > 0$. Then let η be a $\mathrm{Ber}(\varepsilon)$ random variable independent of (Z, A, Y). We consider a variable A_η that takes value A with probability $1 - \varepsilon$ and otherwise flips the variable A with probability ε, that is,

$$A_\eta = A + \eta \pmod 2.$$

This gives us a triple (Z, A_η, Y) that satisfies $\mathbb{E}[A|Z] \in (0, 1)$ and $\mathbb{E}[Y|A, Z] \in (0, 1)$ almost surely by construction, corresponding to condition (iii) from the Theorem. We can also show that the triple satisfies the other two conditions. For instance, to show that condition (i) holds, let S be an arbitrary set such that $S \in \sigma(Z)$.

We will use the fact that any $\sigma(Z)$-measurable random variable V and any random variable U satisfy $\mathbb{E}[\mathbb{E}[U|Z]V] = \mathbb{E}[UV]$. In particular,

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}[f(A_\eta)g(Y)|Z]\mathbb{1}_{Z\in S}\right] &= \mathbb{E}\left[f(A_\eta)g(Y)\mathbb{1}_{Z\in S}\right], (S \in \sigma(Z)) \\
&= \mathbb{E}_{(A,Z,Y)}\left[\mathbb{E}_\eta[f(A_\eta)]g(Y)\mathbb{1}_{Z\in S}\right], (\eta \perp\!\!\!\perp (A, Y, Z)) \\
&= \mathbb{E}_{(A,Z,Y)}\left[\mathbb{E}[\mathbb{E}_\eta[f(A_\eta)]g(Y)|Z]\mathbb{1}_{Z\in S}\right], (S \in \sigma(Z)) \\
&= \mathbb{E}_{(A,Z,Y)}\left[\mathbb{E}[\mathbb{E}_\eta[f(A_\eta)]|Z]\mathbb{E}[g(Y)|Z]\mathbb{1}_{Z\in S}\right], (Y \perp\!\!\!\perp A \mid Z) \\
&= \mathbb{E}_{(A,Z,Y)}\left[\mathbb{E}_\eta[f(A_\eta)]\mathbb{E}[g(Y)|Z]\mathbb{1}_{Z\in S}\right] \\
&= \mathbb{E}\left[f(A_\eta)\mathbb{E}[g(Y)|Z]\mathbb{1}_{Z\in S}\right], (\eta \perp\!\!\!\perp (A, Y, Z)) \\
&= \mathbb{E}\left[\mathbb{E}[f(A_\eta)|Z]\mathbb{E}[g(Y)|Z]\mathbb{1}_{Z\in S}\right],
\end{aligned}$$

where the last step follows since $\mathbb{E}[g(Y)|Z]\mathbb{1}_{Z\in S}$ is Z-measurable.

Because S was arbitrary and both $\mathbb{E}[f(A_\eta)|Z]\mathbb{E}[g(Y)|Z]$ and $\mathbb{E}[f(A_\eta)g(Y)|Z]$ are $\sigma(Z)$-measurable, we can conclude that $\mathbb{E}[f(A_\eta)|Z]\mathbb{E}[g(Y)|Z] = \mathbb{E}[f(A_\eta)g(Y)|Z]$ almost surely so (i) is satisfied. A very similar argument shows that (ii) holds. Therefore, by Theorem 1, A_η is independent of (Z, Y). Then given arbitrary bounded and continuous functions f and g,

$$\mathbb{E}[f(A_\eta)g(Z, Y)] = \mathbb{E}[f(A_\eta)]\mathbb{E}[g(Z, Y)].$$

Using the fact that $A_\eta \to A$ as $\eta \downarrow 0$ in $L^2(\mathbb{P})$, and that $h \mapsto \mathbb{E}[h]$ and $(h, h') \mapsto \mathbb{E}[hh']$ are continuous in $L^2(\mathbb{P})$, we conclude by continuity that

$$\mathbb{E}[f(A)g(Z, Y)] = \mathbb{E}[f(A)]\mathbb{E}[g(Z, Y)].$$

Since f and g were arbitrary, we have in fact shown that A is independent of (Z, Y), as wanted.

Thus, we have succeeded in proving the following refinement: under Theorem 1 assumptions (i) and (ii), if Y cannot be perfectly predicted from any realization of Z, then the random variables A and (Y, Z) must be independent.

Since assumptions (i) and (ii) continue to hold if we condition on $Z \in S$ for any S, we can say further that if Theorem 1 conditions (i) and (ii) hold and P is the set of values of Z from which perfect prediction is not possible, i.e. $\mathrm{Var}(Y|Z) > 0$ then A and Y are independent conditionally on $Z \in P$. ◀

A.2 Proof of Proposition 2

Proof. First we can prove a lemma stating that $S(A)$ is convex. To see this, let ξ be an independent $\mathrm{Ber}(\lambda)$ random variable. Then, by iterating expectations, one sees that

$$\alpha(\hat{y} + \xi(\hat{z} - \hat{y}), A) = \lambda\alpha(\hat{z}, A) + (1 - \lambda)\alpha(\hat{y}, A).$$

Using this convexity, we can prove the proposition. Note that the points $\alpha(\mathbb{1}\{p^* \geq c\}, A)$ that make up the group-A ROC curve of p^* describe the error rates achieved by all cutoff classifiers based on p^*, and so they are in $S(A)$. Meanwhile, since

$$\alpha(1 - \hat{y}, A) = (1, 1) - \alpha(\hat{y}, A),$$

the points $(1, 1) - \alpha(\mathbb{1}\{p^* \geq c\}, A)$ must also be in $S(A)$. This corresponds to the group-A ROC curve of the scores $1 - p^*$. Any point in the convex hull of these two ROC curves can be achieved by randomization as in the aforementioned lemma. For further details and intuition, see Section 4 in Hardt et al. [12]. Note that Hardt et al. choose not to illustrate the feasible region below the main diagonal as it corresponds to classifiers that are worse than random.

To show that *all* attainable error rates belong to this set, we use the convexity of $S(A)$ to note that the support points of $S(A)$ correspond to all classifiers that yield extrema of $\gamma_1\alpha_1(\hat{y}, A) + \gamma_2\alpha_2(\hat{y}, A)$ where (γ_1, γ_2) are arbitrary weights. To describe these support points tractably, we can use the result derived later in the appendix (Proposition 7) that shows that optimal classifications can be chosen to depend on only p^* and A, where $p^* = \mathbb{E}[Y|X, A]$. Thus the extrema of $\gamma \cdot \alpha(\hat{y}, A)$ are achieved by cutoff rules $f(p^*, A) = \mathbb{1}\{p^* \geq c\}$ and $f(p^*, A) = \mathbb{1}\{p^* < c\}$, giving support points

$$\bigcup_{c \in [0,1]} \Big\{\alpha(\mathbb{1}\{p^* \geq c\}, A),\ (1, 1) - \alpha(\mathbb{1}\{p^* \geq c\}, A)\Big\},$$

which as we have shown are contained in $S(A)$. Finally, we use the fact that a convex set containing all of its support points is equal to the convex hull of its support points. ◀

A.3 Proof of Extension of Proposition 3

Proof. Suppose that i holds and call $\hat{p}_f$ the fair score for which $\hat{y} = \mathbb{1}\{\hat{p}_f \geq \bar{p}\}$ satisfies equal error rates. Then since $\hat{p}_f$ is calibrated,

$$\mathbb{P}(Y = 1|\hat{y} = 1, A) = \mathbb{E}[Y|\hat{p}_f \geq \bar{p}, A] = \mathbb{E}[\hat{p}_f|\hat{p}_f \geq \bar{p}, A] \geq \bar{p},$$
$$\mathbb{P}(Y = 1|\hat{y} = 0, A) < \bar{p}.$$

So in addition to satisfying equal error rates, $\hat{y}$ satisfies (9) and (10), which are equivalent to the two conditions in (8). Thus ii is a necessary condition for fairness.

Now we show the converse; ii is also sufficient for fairness. Suppose that ii holds and let $\hat{y}_f$ be a classifier satisfying equal error rates and (8). Choose $\hat{p}(\hat{y}_f, A) = \mathbb{P}(Y = 1|\hat{y}_f, A)$. These scores are calibrated by construction. Also, since they satisfy $\hat{p}(\hat{y}_f = 0, A) < \bar{p}$ and $\hat{p}(\hat{y}_f = 1, A) \geq \bar{p}$, they exactly implement the classifier $\hat{y}_f$ at the cutoff $\bar{p}$. ◀

A.4 Proof of Theorem 4

Proof. Building on the above extension of Proposition 3, it is enough for us to show that the existence of the point $(\alpha_1, \alpha_2) \in S(L) \cap S(H)$ satisfying (11) is equivalent to the following: There exists a classifier $\hat{y}$ satisfying equal error rates and (8).

First note that $S(L) \cap S(H)$ is nonempty, since for example $(0, 0)$ and $(1, 1)$ are points in both $S(L)$ and $S(H)$. So we can consider some arbitrary (α_1, α_2) that is in $S(L) \cap S(H)$ and is therefore implementable by an equal error rate classifier that we call $\hat{y}_e$. We need to show that $\hat{y}_e$ satisfying the conditions in (8) $\forall A$ is equivalent to its corresponding true and false positive rates $(\alpha_1(\hat{y}_e, A), \alpha_2(\hat{y}_e, A))$ satisfying (11) $\forall A$. Recall that the PPV condition in (8) required

$$\mathbb{P}(Y = 1|\hat{y}_e = 1, A) \geq \bar{p}.$$

Applying Bayes' rule to the inequality, we have

$$\begin{aligned}\mathbb{P}(Y = 1|\hat{y}_e = 1, A) &= \frac{\mathbb{P}(\hat{y}_e = 1|Y = 1, A)\mathbb{P}(Y = 1|A)}{\mathbb{P}(\hat{y}_e = 1|A)} \\ &= \frac{\alpha_2(\hat{y}_e, A)\mu_A}{\alpha_2(\hat{y}_e, A)\mu_A + \alpha_1(\hat{y}_e, A)(1 - \mu_A)} \geq \bar{p}.\end{aligned}$$

After algebraic manipulation, the restriction can be written

$$\frac{\alpha_2(\hat{y}_e, A)}{\alpha_1(\hat{y}_e, A)} \geq \frac{\bar{p}(1 - \mu_A)}{(1 - \bar{p})\mu_A} = \frac{\bar{p}}{(1 - \bar{p})\beta_A}.$$

where $\beta_A \equiv \mu_A/(1-\mu_A)$. Therefore $(\alpha_1(\hat{y}_e, A), \alpha_2(\hat{y}_e, A))$ must satisfy the following

$$\frac{\alpha_2(\hat{y}_e, A)}{\alpha_1(\hat{y}_e, A)} \geq \frac{\bar{p}}{(1 - \bar{p})\beta_A}$$

Since $\beta_L < \beta_H$, the condition is more restrictive when $A = L$, giving the first condition in (11). We next similarly transform the NPV condition in (8), recalling it requires $\mathbb{P}(Y = 0|\hat{y} = 0, A) > 1 - \bar{p}$. By Bayes' rule,

$$\begin{aligned}\mathbb{P}(Y = 0|\hat{y} = 0, A) &= \frac{\mathbb{P}(\hat{y} = 0|Y = 0, A)\mathbb{P}(Y = 0|A)}{\mathbb{P}(\hat{y} = 0|A)} \\ &= \frac{(1 - \alpha_1(\hat{y}, A))(1 - \mu_A)}{(1 - \alpha_1(\hat{y}, A))(1 - \mu_A) + (1 - \alpha_2(\hat{y}, A))\mu_A} > 1 - \bar{p}.\end{aligned}$$

After algebraic manipulation, this becomes $\forall A$

$$\frac{(1-\alpha_1(\hat{y},A))}{(1-\alpha_2(\hat{y},A))} > \frac{(1-\bar{p})\beta_A}{\bar{p}}.$$

Since $\beta_H > \beta_L$, the most restrictive case is when $A = H$, giving the second condition in (11).

Note that special attention should be given to the corner solutions. At point $(0,0)$, the first condition in (11) becomes irrelevant and so the second condition in (11) is necessary and sufficient. Meanwhile at $(1,1)$, the second condition in (11) becomes irrelevant so the first condition in (11) is necessary and sufficient. ◀

A.5 Proof of Corollary 5

Proof. Let F and G denote the lines for which the inequalities (11) hold with equality. That is to say, $F, G \subset \mathbb{R}^2$ are given by

$$F = \left\{(\alpha_1,\alpha_2) \in \mathbb{R}^2 \,\middle|\, \frac{\alpha_2}{\alpha_1} = \frac{\bar{p}}{\beta_L(1-\bar{p})}\right\}, \qquad G = \left\{(\alpha_1,\alpha_2) \in \mathbb{R}^2 \,\middle|\, \frac{(1-\alpha_1)}{(1-\alpha_2)} = \frac{\beta_H(1-\bar{p})}{\bar{p}}\right\}$$

The lines intersect at $(\breve{\alpha}_1, \breve{\alpha}_2)$ given by (12). Our proof will rest on a few basic facts: $S(L) \cap S(H)$ is convex, F contains $(0,0)$, G contains $(1,1)$, and both lines have positive slope. First we prove that if $\breve{\alpha}_1 \leq 0$, $\breve{\alpha}_1 \geq 1$, or both ROC curves lie above the intersection $(\breve{\alpha}_1, \breve{\alpha}_2)$, then there exists a point (α_1, α_2) satisfying the feasibility conditions in Theorem 4.

Case I: $0 < \breve{\alpha}_1 < 1$ and $(\breve{\alpha}_1, \breve{\alpha}_2)$ lies below both ROC curves. Note that increasing α_2 slackens both inequalities (11). Thus, if $0 < \breve{\alpha}_1 < 1$ and $(\breve{\alpha}_1, \breve{\alpha}_2)$ lies below both ROC curves, there then exists a point $(\breve{\alpha}_1, \alpha_2)$ with $\alpha_2 > \breve{\alpha}_2$ that lies on the minimum of the two ROC curves, hence in $S(H) \cap S(L)$, and moreover the inequalities (11) hold at $(\breve{\alpha}_1, \alpha_2)$. This is a feasible point.

Case II: $\breve{\alpha}_1 \leq 0$. On the other hand, if $\breve{\alpha}_1 \leq 0$, then in $(0,1) \times \mathbb{R}$ the line F lies strictly above G. Then the point $(0,0) \in S(L) \cap S(H) \cap F$ lies above G, meaning that the second condition in (11) holds and the point is feasible.

Case III: $\breve{\alpha}_1 \geq 1$. If $\breve{\alpha}_1 \geq 1$, then in $(0,1) \times \mathbb{R}$ the line G lies strictly above F. Then the point $(1,1) \in S(L) \cap S(H) \cap G$ lies above F, so the first condition in (11) holds and the point is feasible.

Finally, we prove the converse that if $0 < \breve{\alpha}_1 < 1$ and $\breve{\alpha}_2$ lies above at least one of the ROC curves, then the feasible region is empty. Let the intersection of $S(L) \cap S(H)$ with the half-space above F be denoted by I_F, and the intersection of $S(L) \cap S(H)$ with the half-space above G be denoted by I_G. We need to show that $I_F \cap I_G$ is empty. The argument follows from the convexity of $S(L) \cap S(H)$ and the fact that both F and G have positive slopes. In particular, due to the convexity of $S(L) \cap S(H)$, the positive slope of F, and the fact that $(0,0)$ is in F, we know the line F must intersect the boundary of $S(L) \cap S(H)$ strictly to the left of $\breve{\alpha}_1$. Meanwhile, G must intersect the boundary of $S(L) \cap S(H)$ strictly to the right of $\breve{\alpha}_1$. Thus the rightmost point of I_F lies strictly to the left of the leftmost point of I_G, and the intersection of $S(L) \cap S(H)$ with both half-spaces above F and G must be empty. ◀

A.6 Justification for post-processing p^* in algorithm

First we justify post-processing the Bayes optimal p^* to arrive at the optimal fair $\hat{p}$. To do so we adapt Proposition 5.2 from Hardt et al. [12] to our setting and prove the following

▶ **Proposition 7.** *For any source distribution over (Y, X, A) with Bayes optimal regressor given by $p^*(X, A) = \mathbb{E}[Y|X, A]$ and loss function ℓ, there exists a predictor $\hat{p}(p^*, A)$ such that*

(i) *$\hat{p}$ is an optimal predictor satisfying our fairness properties of calibration and equal error rates. That is, $\mathbb{E}[\ell(\mathbb{1}_{\hat{p}>\bar{p}}, Y)] \leq \mathbb{E}[\ell(\mathbb{1}_{\hat{g}>\bar{p}}, Y)]$ for any $\hat{g}$ that satisfies the properties.*

(ii) *$\hat{p}$ is derived from (p^*, A). In particular, it is a (possibly random) function of the random variables (p^*, A) alone, and is independent of X conditional on (p^*, A).*

Proof. To start, first note that our fairness properties of calibration and equal error rates on a score p and classifications $\mathbb{1}\{p \geq \bar{p}\}$ are "oblivious." That is, they depend only on the joint distribution of (Y, A, p) given the known cutoff $\bar{p}$. We will show that for any arbitrary $\hat{g}$ that satisfies the fairness properties, we can construct a $\hat{p}$ that also satisfies fairness, yields the same expected loss, and is derived from (p^*, A).

Consider an arbitrary $\hat{g} = f(X, A)$ satisfying the fairness properties. We can define $\hat{p}(p^*, A)$ as follows: draw a vector X' independently from the conditional distribution of X given the realized values of p^* and A, and set $\hat{p} = f(X', A)$. Note this $\hat{p}$ satisfies (ii) by construction.

To show that this $\hat{p}$ satisfies the fairness properties and yields the same expected loss as $\hat{g}$, note that since Y is binary with conditional expectation equal to the Bayes optimal p^*, we know Y is independent of X conditional on p^*. Therefore (Y, p^*, X, A) and (Y, p^*, X', A) have the same joint distribution, and so must $(f(X, A), A, Y)$ and $(f(X', A), A, Y)$. Since the fairness properties are oblivious and depend only on these latter joint distributions, then we know that as long as $\hat{g}$ satisfies them then so will $\hat{p}$. Finally, we can deduce that $(Y, \hat{g})$ and $(Y, \hat{p})$ also have the same joint distribution, meaning that (i) is satisfied with equality. ◀

A.7 Our algorithm as a mean-preserving contraction of scores

We observe that a calibrated score derived from another is a mean-preserving contraction. Since the Bayes optimal p^* that serves as input to our algorithm frequently satisfies calibration (see Liu et al. 2019), then our post-processing method can be viewed as finding its smallest mean preserving contraction that achieves equal error rates at the decision-maker's cutoff.

The relationship between calibrated scores related by post-processing is characterized by our proposition below.

▶ **Proposition 8.** *Let p_A be any calibrated score of group A, i.e. satisfying $\mathbb{E}[Y|p_A] = p_A$ for members of A, and let $\hat{p}_A = f(p_A, \zeta)$ be a score post-processed from p_A that is also calibrated, where ζ is independent of Y conditional on p_A. Then, $\hat{p}_A$ is a mean-preserving contraction of p_A, with $p_A = \hat{p}_A + Z$ and $\mathbb{E}[Z|\hat{p}_A] = 0$. Conversely, any $\tilde{p}_A$ that satisfies $p_A = \tilde{p}_A + Z$ with $\mathbb{E}[Z|\tilde{p}_A] = 0$ is calibrated.*

Proof. We first show that $\hat{p}_A$ is a mean-preserving contraction of p_A. To start, note that the post-processed $\hat{p}_A$ is assumed to be calibrated, so $\mathbb{E}[Y|\hat{p}_A] = \hat{p}_A$. Moreover, since $\hat{p}_A = f(p_A, \zeta)$, we have $\sigma(\hat{p}_A) \subseteq \sigma(p_A, \zeta)$. Therefore by the tower property of conditional expectation,

$$\begin{aligned}\hat{p}_A = \mathbb{E}[Y|\hat{p}_A] &= \mathbb{E}[\mathbb{E}[Y|p_A, \zeta]|\hat{p}_A] \\ &= \mathbb{E}[\mathbb{E}[Y|p_A]|\hat{p}_A], \text{(by conditional independence of } \zeta\text{)} \\ &= \mathbb{E}[p_A|\hat{p}_A], \text{(by calibration of } p_A\text{).}\end{aligned}$$

Then $p_A = p_A + (\hat{p}_A - \mathbb{E}[p_A|\hat{p}_A]) = \hat{p}_A + (p_A - \mathbb{E}[p_A|\hat{p}_A])$ where the second term is by construction mean independent of $\hat{p}_A$, so $\hat{p}_A$ is a mean-preserving contraction of p_A.

Now we show that if the score $\tilde{p}_A$ is a mean-preserving contraction of p_A such that $p_A = \tilde{p}_A + Z$ for some Z satisfying $\mathbb{E}(Z|\tilde{p}_A) = 0$, then $\tilde{p}_A$ is calibrated. Observe that

$$\mathbb{E}[p_A|\tilde{p}_A] = \mathbb{E}[\tilde{p}_A + Z|\tilde{p}_A] = \mathbb{E}[\tilde{p}_A|\tilde{p}_A] + \mathbb{E}[Z|\tilde{p}_A] = \tilde{p}_A$$

which is sufficient to show that $\tilde{P}_A$ is calibrated. To see why, recall that p_A is calibrated and note that by the tower property of conditional expectation with $\sigma(\tilde{p}_A) \subseteq \sigma(p_A)$,

$$\mathbb{E}[p_A|\tilde{p}_A] = \mathbb{E}[\mathbb{E}(Y|p_A)|\tilde{p}_A] = \mathbb{E}[Y|\tilde{p}_A].$$ ◀

A.8 Justification for discretizing p^*

Our algorithm uses the discretization of p^* to construct a linear program that maps probability masses from p^* to $\hat{p}$. Note that even if the original p^* is not discrete, it can easily be discretized into N bins by taking $p' = \lfloor Np^* \rfloor / N$. The discretized score will satisfy $|p' - p^*| \leq N^{-1}$ almost surely, so for large values of N, the discretization p' approximates p^* well.

Census TopDown: The Impacts of Differential Privacy on Redistricting

Aloni Cohen ✉
Hariri Institute for Computing and School of Law, Boston University, MA, USA

Moon Duchin ✉
Department of Mathematics, Tufts University, Medford, MA, USA

JN Matthews ✉
Tisch College of Civic Life, Tufts University, Medford, MA, USA

Bhushan Suwal ✉
Tisch College of Civic Life, Tufts University, Medford, MA, USA

Abstract

The 2020 Decennial Census will be released with a new disclosure avoidance system in place, putting *differential privacy* in the spotlight for a wide range of data users. We consider several key applications of Census data in redistricting, developing tools and demonstrations for practitioners who are concerned about the impacts of this new noising algorithm called TopDown. Based on a close look at reconstructed Texas data, we find reassuring evidence that TopDown will not threaten the ability to produce districts with tolerable population balance or to detect signals of racial polarization for Voting Rights Act enforcement.

2012 ACM Subject Classification Security and privacy; Applied computing → Law; Applied computing → Voting / election technologies

Keywords and phrases Census, TopDown, differential privacy, redistricting, Voting Rights Act

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.5

Supplementary Material *Text (Documentation)*: `https://mggg.org/DP`
Software (Source Code): `https://github.com/mggg/census-diff-privacy`
archived at `swh:1:dir:aeb9390dfc99f5bc755546b21a1e9ddaecabb65e`

Funding This project was supported on NSF OIA-1937095 (Convergence Accelerator) and by a grant from the Alfred P. Sloan Foundation.
Aloni Cohen: NSF CNS-1414119; NSF CNS-1915763; DARPA HR00112020021.
Moon Duchin: NSF DMS-2005512

Acknowledgements Authors are listed alphabetically. We thank Denis Kazakov, Mark Hansen, and Peter Wayner. Kazakov developed the reconstruction algorithm as a member of Hansen's research group. Wayner guided our deployment of TopDown in AWS and was an invaluable team member for the technical report. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our funders.

1 Introduction

A new disclosure avoidance system is coming to the Census: the 2020 Decennial Census releases will use an algorithm called TopDown to protect the data from increasingly feasible *reconstruction attacks* [2]. Census data is structured in a nesting sequence of geographic units covering the whole country, from nation at the top to small *census blocks* at the bottom. TopDown starts by setting a *privacy budget* $\varepsilon > 0$ which is allocated to the levels of a designated hierarchy, then adding noise at each level in a *differentially private* way [12]. When $\varepsilon \to \infty$, the data alterations vanish, while $\varepsilon \to 0$ yields pure noise with no fidelity to the input data. The algorithm continues with a post-processing step that leaves an output dataset that is designed to be suitable for public use.

Redistricting is the process of dividing a polity into territorially delimited pieces in which elections will be conducted. The Census has a special release – named the PL 94-171 after the law that requires it – that reports the number of residents in every geographic unit in the country by race, ethnicity, and the number of voting-age residents [9]. The 2020 release is slated to occur by September 2021, after which many thousands of district lines will be redrawn: not only U.S. Congressional districts, but those for state legislatures, county commissions, city councils, and many more.

Many user groups have expressed concerns about the effects of differential privacy on redistricting. They largely but not exclusively concern two issues. First, "One Person, One Vote" case law calls for balancing population across the electoral districts in a jurisdiction, whether small like city council districts or large like congressional districts. Most states balance congressional districts to within one person based on Census counts. Second, the most reliable legal tool against gerrymandering has been the Voting Rights Act of 1965 (VRA), which requires a demonstration of racially polarized voting (RPV). This RPV analysis is typically performed by statistical techniques that infer voting by race from precinct-level returns. Many voting rights advocates worry that noising of Census data will confuse population balancing practices, and others worry that it will attenuate RPV signals, making it harder to press valid claims.

The Census Bureau has been commendably transparent about the development of TopDown, making working code publicly available along with documentation and research papers describing the algorithm. The complexity of the algorithm makes it extremely difficult to study analytically, so many people have sought to run it on realistic data. However, since person-level Census data remain confidential for 72 years after collection, detailed input data for TopDown is not public. Data users who would like to understand its impacts are left with two options: decades-old data or a limited demonstration data product.

In this paper, we get around the empirical obstacle by use of reconstructed block-level 2010 microdata for the state of Texas, and we try to understand the algorithm through theoretical analysis of a much-simplified toy algorithm, ToyDown, that retains the two-stage, top-down structure of TopDown but is much easier to analyze symbolically. We investigate three questions about the count discrepancies created by TopDown in units of census geography and "off-spine" aggregations like districts and precincts.

Hierarchical budget allocation. We derive easy-to-evaluate expressions for ToyDown errors as a function of the privacy budget allocation. Error at higher levels of the geographic hierarchy impacts lower-level counts with a significant discount, suggesting that bottom-heavy allocations may be optimal for accuracy on small geographies. This is consistent with the small-district errors in our experiments with TopDown. For larger districts, a tract-heavy allocation gives greatest accuracy. Equal allocation over the levels is a strong performer in both cases, making this a good choice from the point of view of multi-scale redistricting.

District construction. From there, we create further tests to study the impacts of district design. We compare hierarchically greedy to geometrically greedy district-generation schemes, where the former attempt to keep large units whole and the latter attempt to build districts with short boundaries. We find that the ToyDown model gives errors very closely keyed to the fragmentation of the hierarchy, but that spatial factors damp out the primary role of fragmentation in the shift to the TopDown setting.

Robustness of linear regression. Finally, we consider the unweighted linear regressions commonly used to assess racial polarization in voting rights cases. We find that the noise from both ToyDown and TopDown introduces an attenuation bias that seems alarming at first. However, unweighted linear regression on precincts is already vulnerable to major skews imposed by the inclusion of very small precincts. For any reasonable way of counteracting that – trimming out the tiny precincts or weighting the regression by the number of votes cast – the instability introduced by ToyDown and TopDown all but vanishes.

Our investigation is set up to answer questions about the status quo workflow in redistricting. As usual with studies of differential privacy, a finding that DP unsettles the current practices might lead us to call to refine the way it is applied, but might equally lead us to interrogate the traditional practices and seek next-generation methods for redistricting. In particular, it is clear that the practice of *one-person* population deviation across districts was never reasonably justified by the accuracy of Census data nor required by law, and the adoption of differential privacy might give redistricters occasion to reconsider that practice. We make a similar observation about the way that racially polarized voting analysis is commonly performed in expert reports. On the other hand, by focusing on decisions still to be announced like the privacy budget and its allocation over the hierarchy, we are able to make recommendations that can assist the Bureau in protecting privacy while attending to the important concerns of user groups.

2 Background on Census and redistricting

2.1 The structure of Census data and the redistricting data products

Every ten years the U.S. Census Bureau attempts a comprehensive collection of person-level data – called *microdata* – from every household in the country. The microdata are confidential, and are only published in aggregated tables subject to disclosure avoidance controls. The Decennial Census records information on the sex, age, race, and ethnicity for each member of each household, using categories set by the Office of Management and Budget [8]. The 2020 Census used six primary racial categories: White, Black, American Indian, Asian, Native Hawaiian/Pacific Islander, and Some Other Race. An individual can select these in any combination but must choose at least one, creating $2^6 - 1 = 63$ possible choices of race. Separately, *ethnicity* is represented as a binary choice of Hispanic/Latino or not.

The 2010 Census divided the nation into over 11 million small units called *census blocks* which nest in larger geographies in a six-level "central spine": nation – state – county – tract – block group – block. Counts of different types are provided with respect to these geographies. This tabular data is then used in an enormous range of official capacities, from the apportionment of seats in the U.S. House of Representatives to the allocation of many streams of federal and state funding. The redistricting (PL 94-171) data includes four such tables: H1, a table of housing units whose types are occupied/vacant; and four tables of population, P1 (63 races), P2 (Hispanic, and 63 races of non-Hispanic population), and P3/P4 (same as P1/P2 but for voting age population). Each table can be thought of as a *histogram*, with each included type constituting one histogram *bin*. For instance, in table P1 there is 1 person in the t =White+Asian bin in the Middlesex County, MA, block numbered 31021002.

Treating the 2010 tables as accurate, it is easy to infer information not explicitly presented in the tables. For instance, the same bin in the P3 table (race for voting age population) also has a count of 1, implying that there are no White+Asian people under 18 years old in block 31021002. This is the beginning of a *reconstruction* process that would enable an attacker, in principle, to learn much of the person-level microdata behind the aggregate releases.

2.2 Disclosure avoidance

Title 13 of the U.S. Code requires the Bureau to take measures to protect the privacy of respondents' data [1]. In the 2010 Census, this was largely achieved by an ad hoc mechanism called *data swapping*: a Bureau employee manually swapped data between small census blocks to thwart re-identification. In 2020, swapping is no longer considered adequate to protect against more sophisticated (but mathematically straightforward) data attacks that seek to reconstruct the individual microdata. An internal Census Bureau study concluded that data swapping was unacceptably vulnerable: Census staff were able to reconstruct the 2010 Census responses of – and correctly reidentify – tens of millions of people.

With the reconstruction/reidentification threat in mind, the Bureau has developed an algorithm called TopDown [2], which begins with a noising step that is *differentially private*, following a mathematical formalism that provides rigorous guarantees against information disclosure [12]. Differentially private algorithms obey a quantifiable limit to how much the output can depend on an individual record in the input. The relationship of output to input is specified by a tuneable parameter, ε, often called the *privacy budget*. When $\varepsilon \to \infty$, the output approaches equality to the input (high risk of disclosure). When $\varepsilon \to 0$, the output bears no resemblance to the input whatsoever (no risk of disclosure). Like a fiscal budget, the privacy budget can be allocated until it is fully spent, in this case by spending parts of the budget on particular queries and on levels of the hierarchy.

TopDown takes an individual-level table of census data and creates a "synthetic" dataset that will be used in its place to generate the PL 94-171 tables. It can be thought of as taking as input a histogram with a bin for each person type (i.e., a combination of race, sex, ethnicity, etc.) and outputting an altered version of the same histogram. It proceeds in two stages. First, it privatizes the input histogram counts: it adds enough random noise to get the required level of differential privacy (according to the budget ε). At this stage, it also allocates a portion of the total privacy budget for generating additional noisy histograms of data of particular importance to the Census Bureau. Second, TopDown does post-processing on the noisy histograms to satisfy a handful of additional plausibility constraints. Among other things, post-processing ensures that the resulting histograms contain only non-negative integers, are self-consistent, and agree with the raw input data on a handful of *invariants* (e.g., total state population).

The overall privacy guarantees of TopDown are poorly understood. In this paper, we design a simpler cousin of TopDown nicknamed ToyDown and we explore the properties of both ToyDown and TopDown, primarily focusing on reconstructed Texas data from 2010.

2.3 The use of Census products for redistricting

The PL 94-171 tables are the authoritative source of data for the purposes of apportionment to the U.S. House of Representatives, and with a very small number of exceptions also for within-state legislative apportionment. The most famous use of population counts is to decide how many members of the 435-seat House of Representatives are assigned to each state. In "One person, one vote" jurisprudence initiated in the *Reynolds v. Sims* case of 1964, balancing Census population is required not only for Congressional districts within a state but also for districts that elect to a state legislature, a county commission, a city council or school board, and so on [17, 18, 3].

Today, the Congressional districts within a state usually balance total population extremely tightly: each of Alabama's seven Congressional districts drawn after the 2010 Census has a total population of either 682,819 or 682,820 according to official definitions of districts and the Table P1 count, while Massachusetts districts all have a population of 727,514 or 727,515. Astonishingly, though no official rule demands it, more than half of the states maintain this "zero-balancing" practice (no more than one person deviation) for Congressional districts [16]. This ingrained habit of zero-balancing districts to protect from the possibility of a malapportionment challenge is the first source of worry in the redistricting sphere. If disclosure avoidance practices introduce some systematic bias – say by creating significant net redistribution towards rural and away from urban areas – then it becomes hard to control overall malapportionment, which could in principle trigger constitutional scrutiny. In the end, redistricters may not care very much how many people live in a single census block, but it could be quite important to have good accuracy at the level of a district.

The second major locus of concern for redistricting practitioners is the enforcement of the Voting Rights Act (VRA). Here, histogram data is used to estimate the share of voting age population held by members of minority racial and ethnic groups. Voting rights attorneys must start by satisfying three threshold tests without which no suit can go forward.

- **Gingles 1**: the first "Gingles factor" in VRA liability is satisfied by creating a demonstration district where the minority group makes up over 50% of the voting age population.
- **Gingles 2-3**: the voting patterns in the disputed area must display *racial polarization*. The minority population is shown to be cohesive in its candidates of choice, and bloc voting by the majority prevents these candidates from being elected. In practice, inference techniques like linear regression or so-called "ecological inference" are used to estimate voting preferences by race.

Since the VRA has been a powerful tool against gerrymandering for over 50 years, many worry that even where the raw data would clear the Gingles preconditions, the noised data will tend towards uniformity – blocking deserving plaintiffs from a cause of action.

3 Census TopDown and ToyDown

3.1 Setup and notation

For the Census application, the data universe is a set of *types*: for instance, the redistricting data (the PL 94-171) has the types $T = T_R \times T_E \times T_{VA} \times T_H$, where T_R is the set of 63 races, T_E is binary for ethnicity (Hispanic or not), T_A is binary for age (voting age or not), and T_H is the set of housing types. (The fuller decennial Census data has more types.)

A *hierarchy* H is a rooted tree of some depth d, so that every leaf has distance $\leq d-1$ from the root. We will usually assume the hierarchy has uniform depth, so that every leaf is exactly $d-1$ away from the root. For node $h \in H$, let $n(h) \in \mathbb{N}$ be the number of children of h in the tree, and let $\ell(h)$ be the level of node h. A hierarchy is called *homogeneous* if each node at level ℓ has the same number of children, denoted n_ℓ. Let H_ℓ denote the set of nodes at level ℓ, so that the set of leaves is H_d in the uniform-depth case. Label the root of the tree $h = 1$. We adopt an indexing of the tree and refer to the ith child of h as h_i; the parent of any non-root node h is denoted $\hat{h}$. In Census data, the hierarchy represents the large and complicated set of nested geographical units, from the nation at the root down to the census blocks at the leaves. The standard hierarchy has the six levels (nation – state – county – tract – block group – block) described above.

We associate with hierarchy H and types T a set of *counts* $A_{H,T} = \{a_{h,t} \in \mathbb{N}\}_{h \in H, t \in T}$, where $a_{h,t}$ is the population of type t in unit h of census geography. We say $A_{H,T}$ is *hierarchically consistent* if the counts add up correctly: for every non-leaf h and every t, we require $a_{h,t} = \sum_{i \in [n(h)]} a_{h_i,t}$. For a singleton T, we write $A_H = \{a_h\}$. We set an *allocation* $(\varepsilon_1, \ldots, \varepsilon_d)$ breaking down the privacy budget $\varepsilon = \sum \varepsilon_i$ to the different levels of the hierarchy.

Our *queries* will always be counting queries, so that for instance $q_{F,44}(h)$ returns the number of 44-year-old females in geographic unit h. This particular query is part of a "sex by age" *histogram* $Q_{sex,age} = \{q_{s,a} : s \in T_S, a \in T_A\}$, which partitions T into *bins* by sex and age. In this language, $q_{F,44}$ is a bin of the sex-by-age histogram. By slight abuse of notation, we will use the same terminology for the queries and their outputs, so that the histogram can be thought of as the collection of queries or the collection of counts. Similarly, the "voting age by ethnicity by race" histogram consists of a query for each combination of the $2 \times 2 \times 63$ possible combinations of the three attributes.

3.2 ToyDown and TopDown

The Bureau's TopDown and our simplified ToyDown are both algorithms for releasing privatized population counts for every $h \in H$. That is, these algorithms protect privacy by noising the data histograms. TopDown releases not just total population counts, but counts by type. We will define *single-attribute* and *multi-attribute* versions of ToyDown that noise A_H and $A_{H,T}$, respectively, where consistency must hold for each type t.

TopDown and ToyDown share the same two-stage structure. Starting with hierarchically consistent raw counts a, the *noising stage* generates differentially private counts $\widehat{a}$. The *post-processing stage* solves a constrained optimization problem to find noisy counts α that are close to the $\widehat{a}$ values while satisfying hierarchical consistency and other requirements. TopDown is named after the iterative approach to post-processing: one geographic level at a time, starting at the top (nation) and working down to the leaves (blocks). We sketch the noising and post-processing here, and we describe them in Appendix A in more detail.

The simple ToyDown model can be run in a single-attribute version (only counts A_H), a multi-attribute version (counts by type $A_{H,T}$), or in multi-attribute form enforcing non-negativity. The single-attribute version is easy to describe: level by level, random noise values are selected from a Laplace distribution with scale $1/\varepsilon_\ell$ and added to each count, replacing each a_h with $\widehat{a}_h = a_h + L_h$. Then, working from top to bottom, the noisy $\widehat{a}_h$ are replaced with the closest possible real numbers α_h satisfying hierarchical consistency. Multi-attribute ToyDown is defined analogously, but using $A_{H,T}$ instead of A_H and requiring hierarchical consistency within each type $t \in T$. Non-negative ToyDown adds the inequality requirement that $\alpha_h \geq 0$.

TopDown is structurally similar but much more complex, with more kinds of privatized counts in the noising stage and a great many more constraints in the post-processing stage, including integrality. The privatized counts computed by TopDown are specified by a collection of histograms (or complex queries) called a *workload* W. For each bin of each histogram in the workload and for each node h in the geographic hierarchy, TopDown adds geometric noise to the count. The post-processing step finds the closest integer point that satisfies the requirements given by hierarchical consistency, non-negativity, as well as additional conditions given as invariants and structural inequalities. For example, any block with zero households in the raw counts must have zero households and zero population in the output adjusted counts. Together, the invariants, structural inequalities, integrality, and non-negativity make this optimization problem very hard. The problem is NP-hard in the worst case and TopDown cannot always find a feasible solution. There is a sophisticated secondary algorithm for finding approximate solutions that is beyond the scope of this paper.

ToyDown is simple enough that solutions can often be obtained symbolically. ToyDown simplifies the noising stage by fixing the workload to be the detailed workload partition $Q_{detailed} = \{\{t\}\}_{t \in T}$ consisting of all singleton sets and using the continuous Laplace Mechanism instead of the discrete Geometric Mechanism. It simplifies the post-processing stage by dropping invariants, structural inequalities, integrality, and non-negativity. When negative answers are permitted, multi-attribute ToyDown is equivalent to executing $|T|$ independent instances of single-attribute ToyDown on inputs $A_{H,t} = \{a_{h,t}\}_{h \in H}$ for each $t \in T$. As a result, many of our analytical results for single-attribute ToyDown extend straightforwardly to multi-attribute ToyDown (allowing negative answers) by scaling by a factor of $|T|$ in appropriate places.

4 Methods

We use both analytical and empirical techniques in this work. This section describes our high-level empirical approach: what algorithms and raw data we used and how we used them. See Appendix B for more details. We repeatedly ran TopDown and ToyDown in various configurations on a reconstructed person-level Texas dataset created by applying a reconstruction technique to the block-level data from the 2010 Census, following [15] based on [11]. The reconstructed microdata records – obtained from collaborators – contain block-level sex, age, ethnicity, and race information consistent with a collection of tables from 2010 Census Summary File 1.

We executed 16 runs of TopDown with each of 20 different allocations of the privacy budget across the five lower levels of the national census geographic hierarchy: $\varepsilon = \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6$. The 20 allocations consist of five different splits across the levels (Table 1) for each of four total budgets $\varepsilon \in \{0.25, 0.5, 1.0, 2.0\}$. TopDown operates on the six-level Census hierarchy and requires specifying ε_1. In our experiments, we ran TopDown with a fixed total privacy budget $\varepsilon_{total} = 10$, with $\varepsilon_1 = 10 - \varepsilon$. Because the nation-level budget is so much higher than the lower level budgets, we omit further discussion of it. The TopDown workload was modeled after the workload used in the 2018 End-to-End test release, omitting household invariants and queries.

We also ran three variants of ToyDown (single-attribute, multi-attribute, and non-negative) on a simplified version of the same data 2010 data. We executed 16 runs of each variant with each of five different splits of the privacy budget across the five lower levels of the census geographic hierarchy (Table 1), fixing the total budget for those five levels at $\varepsilon = 1$. The data was derived from the reconstructed Texas data simplified to include only seven distinct types: one for the total Hispanic population and one for each of six subgroups of the non-Hispanic population based on race (White; Black; American Indian; Asian; Native Hawaiian/Pacific Islander; and Some Other Race or multiple races). Post-processing for single-attribute ToyDown was implemented in NumPy, while post-processing for multi-attribute and non-negative ToyDown used a Gurobi solver.

5 Hierarchical budget allocation

The relationship of the hierarchical allocation $(\varepsilon_1, \ldots, \varepsilon_d)$ to various measures of output accuracy is not obvious. On one hand, it might seem that higher values of ε_d (the block-level budget) will best promote accuracy at the block level, for a fixed ε. But on the other hand, imposing hierarchical consistency forces lower levels to be consistent with the totals at higher levels, which means that noise at higher levels can trickle down to lower levels. These competing effects create tradeoffs that are hard to balance without further analysis.

Table 1 Names of designated budget splits used in ToyDown and TopDown runs below, each with a budget of $\varepsilon_1 = 9$ on the nation and a total of 1 allocated below the national level.

	state	county	tract	BG	block
Split name	ε_2	ε_3	ε_4	ε_5	ε_6
equal	0.2	0.2	0.2	0.2	0.2
state-heavy	0.5	0.25	0.083	0.083	0.083
tract-heavy	0.083	0.167	0.5	0.167	0.083
BG-heavy	0.083	0.083	0.167	0.5	0.167
block-heavy	0.083	0.083	0.083	0.25	0.5

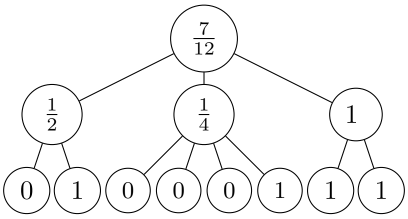


Figure 1 A district in a three-level hierarchy. The 0/1 weight of a leaf indicates its membership in the district; each non-leaf weight is the average of the node's children.

5.1 ToyDown error expressions

▶ **Definition 1** (District, weights, error). *A* district $D \subseteq H_d$ *is a subset of the leaves (blocks) of the hierarchy H. For hierarchy H, a district D induces* weights $w_h \in [0,1]$ *on the hierarchy nodes, defined recursively as follows:*

- *For each leaf $h \in H_d$, let $w_h = 1$ if $h \in D$ and $w_h = 0$ otherwise.*
- *For $\ell \leq d-1$ and $h \in H_\ell$, let $w_h = \frac{1}{n(h)} \cdot \sum_{i \in [n(h)]} w_{h_i}$ be the average of the weights of the children.*

In a homogeneous hierarchy, we can observe that each w_h equals the fraction of the leaves descended from h that belong to D. In particular, the root weight is $w_1 = |D|/|H_d| = 1/k$ if there are k districts of equal population made from nodes of equal population.

For node $h \in H$, we record the *error* $E_h = \alpha_h - a_h$ introduced by ToyDown to the count a_h. The total error over district D is $E_D = \sum_{h \in D} E_h$. Let $\hat{h}$ denote the parent of node h.

▶ **Theorem 2** (Error expressions). *$E_1 = L_1$. For $\ell \in \{2, \ldots, d\}$ and non-root node $h_i \in H_\ell$, and for every district D with associated weights w_h on the nodes,*

$$E_{h_i} = L_{h_i} + \frac{1}{n(h)} \left(E_h - \sum_{j \in [n(h)]} L_{h_j} \right), \qquad E_D = w_1 L_1 + \sum_{h \in H \setminus \{1\}} (w_h - w_{\hat{h}}) L_h. \tag{1}$$

We make several observations. First, our intuition that error at higher levels trickles down to lower levels is correct, but this effect is rather weak. The error at a child h_i is determined by the parent error E_h discounted by the degree $n(h)$, the number of siblings. This suggests that placing more budget at level ℓ is an efficient way to secure accuracy at that level, until a fairly extreme level of error at higher levels overwhelms the degree-based "discount."

Second, because the L_h are all independent random variables with $\mathbb{E}(L_h) = 0$ and $\text{Var}(L_h) = 8/\varepsilon^2_{\ell(h)}$, the theorem provides the following expression for variance that we use repeatedly.

▶ **Corollary 3** (Error expectation and variance). *For all $D \subseteq H_d$ and associated weights w_h, the expected error and error variance produced by ToyDown satisfy $\mathbb{E}(E_D) = 0$ and*

$$\text{Var}(E_D) = \frac{8w_1^2}{\varepsilon_1^2} + \sum_{\ell=2}^{d} \left(\frac{8}{\varepsilon_\ell^2} \cdot \sum_{h \in H_\ell} (w_h - w_{\hat{h}})^2 \right). \tag{2}$$

Third, we get a more explicit expression if restricting to homogeneous hierarchies H. Consider the case of a singleton district $\{h\}$ made of a single census block $h \in H_d$.

▶ **Corollary 4** (Error variance, homogeneous case). *The ToyDown error for a single block $h \in H_d$ satisfies*

$$\mathrm{Var}(E_h) = \frac{8}{\varepsilon_1^2 (n_1 \cdots n_{d-1})^2} + \sum_{\ell=2}^{d} \frac{8 n_{\ell-1}(n_{\ell-1} - 1)}{\varepsilon_\ell^2 (n_{\ell-1} \cdots n_{d-1})^2}. \tag{3}$$

Figure 2 plots this expression for various ways of splitting a total privacy budget of $\varepsilon = 1$ across a three-level hierarchy with $n_1 = n_2 = 10$. The minimum of $f(x_1, \dots, x_d) = \sum_{\ell=1}^{d} a_\ell / x_\ell^2$ subject to $\sum_\ell x_\ell = \varepsilon$ and $x_\ell \geq 0$ is achieved at $x_\ell = \varepsilon a_\ell^{1/3} / \sum_i a_i^{1/3}$ for all ℓ. For the example in Figure 2, the minimum-variance split is $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0.038, 0.171, 0.791)$ with variance 14.52. (See accompanying CoLab notebook.) One important note in interpreting Figure 2 is that these variance numbers are absolute and don't depend on knowing population counts for the nodes of the hierarchy. They are simply based on sampling Laplace noise with the given parameters. If a variance of about 15 in the bottom-level counts is too high to be tolerated in an application, one would have to increase ε to achieve lower variance.

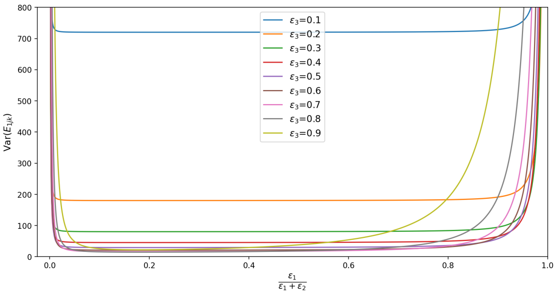


Figure 2 ToyDown error variance for a leaf node in the three-level hierarchy with $n_1 = n_2 = 10$ and $\varepsilon = 1$. The curves show varying ε_3 (colors) and the relative balance of ε_1 and ε_2 (x-axis).

Table 2 L^1 error measurements from selected TopDown runs on reconstructed Texas data. The allocation $(\varepsilon_1, \dots, \varepsilon_6)$ goes from the nation $\ell = 1$ down to census blocks at $\ell = 6$.

ε	Allocation	L^1 error
1.0	(.16, .16, .16, .16, .16, .2)	0.03
1.0	(.2, .16, .16, .16, .16, .16)	0.03
1.0	(.1, .1, .1, .1, .1, .5)	0.02
1.0	(.02, .02, .02, .02, .02, .9)	0.03
1.0	(.66, .30, .01, .01, .01, .01)	0.09

5.2 Empirical error experiments in TopDown

Next, we move to TopDown, which requires the use of input data. First, using reconstructed 2010 Texas data, we varied the relative allocation vector and the total ε, then measured the effects with an L^1 error metric included in the Census code [5]. This is a measure of block-level error: it adds the magnitudes of changes in the bins, then divides by twice the total population in the histogram.

Table 2 reports a small selection of the 100+ different scenarios explored. In general, the lowest error outcomes were observed in a few scenarios: when the budget was distributed near-equally to the levels of the hierarchy, and when half of the available budget was placed at the bottom level – beyond $\varepsilon_d = \varepsilon/2$, further bottom-weighting gave diminishing returns in block-level accuracy.

But a budget allocation that produces small block-level errors may not produce small errors for *districts*, depending on the degree of cancellation or correlation. Next, we use random district generation to understand the effects of off-spine aggregation. In particular, we employ the Markov chain sampling algorithm called *recombination* (or ReCom), which runs an elementary move that fuses two neighboring districts and re-partitions the double-district by a random balanced cut to a random spanning tree [10].

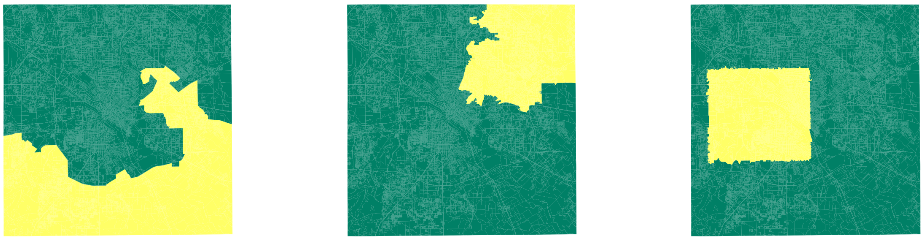


Figure 3 Three sample districts (yellow) in Dallas County, each within two percent of the ideal population for $k = 4$ districts. These are drawn by tract ReCom, block ReCom, and a square-favoring algorithm, respectively.

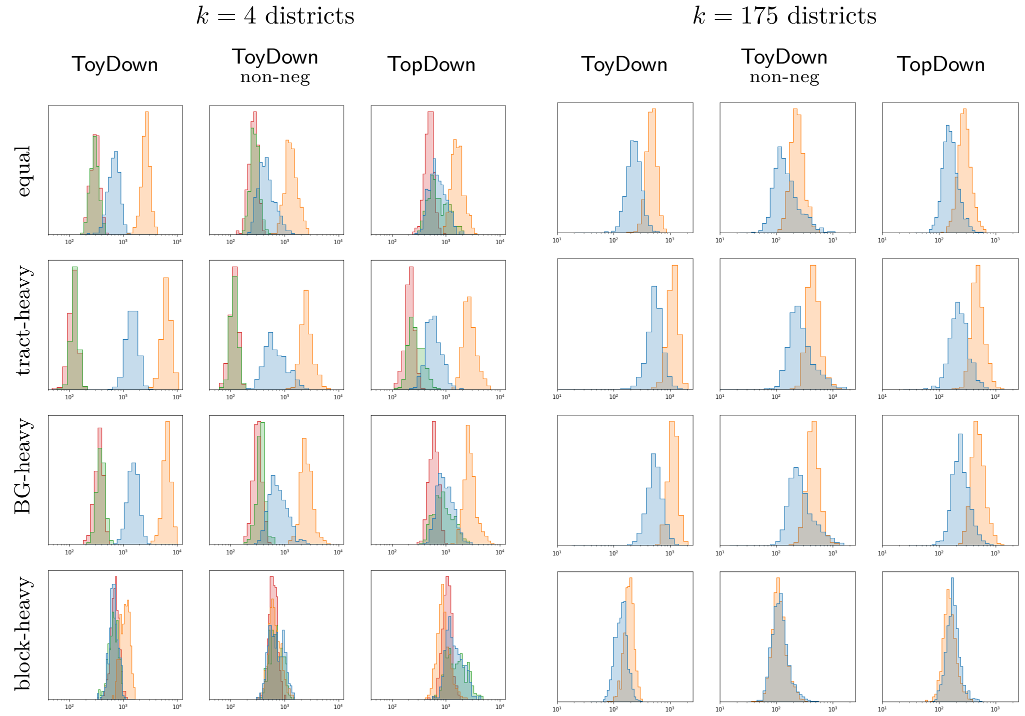
We begin with county commission districts in Dallas County, where $k = 4$. Since the 2010 population of Dallas County was roughly 2.4 million, each district will have roughly 600,000 people, making them nearly as big as congressional districts and much larger than tracts. We also include divisions of the county into $k = 175$ districts of between 13,000 and 14,000 people each for a small-district comparison. Figure 4 plots the data from our experiments on a logarithmic scale. Each histogram displays 400 values, one for each district drawn by the specified district-drawing algorithm; each value is the mean observed district-level population error magnitude over 16 executions of the specified hierarchical noising algorithm using the specified budget allocation.

First, consider two unrealistic forms of district-generation: tract Disconn (red) and block Disconn (orange), which randomly choose units of the specified type until assembling a collection with the appropriate population. These are unrealistic because they do not form connected districts; here, they are used to illustrate the effects of aggregation, neglecting spatial factors entirely. We see in Figure 4 that block-based methods generate hugely more error than tract-based methods, except if the budget allocation is concentrated at the bottom of the hierarchy. The effect is stronger for ToyDown (in keeping with Theorem 2), but is easily observed for TopDown as well.

We compare that with the more realistic district-generation algorithm block ReCom (blue), which builds compact and connected districts out of block units. This tends to give error levels in between the extremes set by the other two. Likewise, tract ReCom (green) builds compact and connected districts from tracts. One reasonable mechanism by which ReCom has much lower error than Disconn is that ReCom districts will tend to have higher "hierarchical integrity," keeping higher-level units whole just by virtue of being connected and plump. The interior of ReCom districts will thus contain many whole block groups and tracts. Near the boundary, block groups and tracts are more fragmented, leaving the corresponding block-level errors uncancelled. These fragmentation ideas are explored more fully in Section 6 and some sample districts are depicted here.

The cancellation effect is significant: in most experiments, the error level for ReCom districts is much closer to that of tract Disconn than block Disconn (recall the data is plotted on a logarithmic scale). Overall, drawing districts out of larger pieces (e.g., using tract Disconn instead of ReCom, or ReCom instead of block Disconn) lowers error magnitude significantly in the best case and has little or no effect in the worst case.

Although tract ReCom and tract Disconn behave very similarly under ToyDown, the compact districts perform noticeably worse than their disconnected relatives once we pass to the full complexity of TopDown. At first this seems puzzling, because compact and



Green: tract ReCom, Red: tract Disconn, Blue: block ReCom, Orange: block Disconn

Figure 4 These histograms show district-level error on a log scale for various combinations of budget splits (rows), district-drawing algorithms (colors), and noising algorithms (columns). We include both large districts and small districts, dividing the county into $k = 4$ and $k = 175$ equal parts. Each histogram displays 400 values, one for each district drawn by the specified algorithm, plotting the mean observed district-level population error magnitude over 16 executions of the noising algorithm using the specified budget allocation.

connected districts are being punished by the geography-aware TopDown. But the reason for this is apparent on further reflection: *spatial autocorrelation* is causing the post-processing corrections to move nearby tracts in the same direction, impeding the cancellation that makes counts usually more accurate on larger geographies.

In the end, the story that emerges from these investigations is that, with full TopDown, the best accuracy that can be observed for large districts occurs when they are made from whole tracts and the allocation is tract-heavy; an equal split is not much worse. For districts with population around 13,000, $\varepsilon = 1$ noising creates errors in the low hundreds for compact, connected districts, with the best performance for block-heavy allocations. Again, an equal split is not much worse, suggesting that this might be a good policy choice for accuracy in districts across many scales.

6 Geometrically compact vs hierarchically greedy districts

The analysis above suggests that the district-level error E_D will depend not only on the randomness of the noising algorithms, but also on the geometry of D and the structure of H. This section studies the hypothesis that districts that disrespect the geographical hierarchy will tend to have higher error magnitude. This section defines the *fragmentation score*, relates a district's fragmentation score to its error variance under ToyDown, and compares the fragmentation of two simple district-drawing algorithms on homogeneous hierarchies and simple geographies. Ultimately, we find that the explanatory value of the fragmentation score decays as we move to more realistic deployment of TopDown. This discrepancy raises important questions for future study: Which of the many additional features of TopDown attenuates the fragmentation–variance relationship?

We define a score intended to capture the contribution to $\mathrm{Var}(E_D)$ of the shape of the district with respect to the hierarchy. Recall that $\hat{h}$ denotes the parent of node h.

▶ **Definition 5** (Fragmentation score). *For $D \subseteq H_d$, let* $\mathsf{Frag}(D) = \sum_{h \in H} (w_h - w_{\hat{h}})^2$.

Because weights are in $[0, 1]$, the score obeys $0 \le \mathsf{Frag}(D) < |H|$ for all districts, with higher scores indicating the presence of more units that are only partially included in D.

This fragmentation score is reverse-engineered from the expression for the variance of district-level population errors when using ToyDown with privacy divided equally across levels of the hierarchy (Corollary 3): namely, $\mathrm{Var}(E_D) = \frac{8d^2}{\varepsilon^2}\left(w_1^2 + \mathsf{Frag}(D)\right)$. When the district D itself is a random variable sampled from some distribution, the expected fragmentation $\mathbb{E}(\mathsf{Frag}(D))$ is similarly related to $\mathrm{Var}(E_D)$. Namely, using the law of total variation, when each level gets ε/d privacy budget:

$$\mathrm{Var}(E_D) = \mathbb{E}\left(\mathrm{Var}(E_D|D)\right) + \mathrm{Var}\left(\mathbb{E}(E_D|D)\right) = \mathbb{E}(\mathrm{Var}(E_D|D)) = \frac{8d^2}{\varepsilon^2}(\mathbb{E}(\mathsf{Frag}(D)) + \mathbb{E}(w_1^2)).$$

When ε is allocated unequally across levels, as for the other splits in Table 1, the simple analytical relationship between the fragmentation score and the error variance breaks down.

Observe that a hierarchy H does not capture all of the geometry relevant to district drawing. In particular, H does not directly encode any information about block adjacency, and therefore we can't detect from H that a district is contiguous. For algorithms to generate contiguous districts, we need to make use of the plane geometry associated to H. We restrict our attention to the simplest case: homogeneous hierarchies (where every node on level $\ell < d$ has exactly n_ℓ children) and *square tilings.* (where each unit on level ℓ is a square and has n_ℓ children that cover it with a $\sqrt{n_\ell} \times \sqrt{n_\ell}$ grid tiling).

We analyze the fragmentation score for two simple district-drawing algorithms (see Appendix C). The Greedy algorithm builds a district from the largest subtrees possible, only subdividing a subtree when necessary. It takes as input H and $k \in \mathbb{N}$ and returns a district of size $N = \lfloor |H_d|/k \rfloor$, assembled by starting with the largest available units at random and adding units that are adjacent in the labeling sequence without passing size N, then allowing one partial unit, and so on recursively at lower levels. Observe that Greedy depends only on the hierarchy H. The Square algorithm takes as input a square, homogeneous hierarchy H and $k \in \mathbb{N}$ such that the district size is a perfect square, $|D| = |H_d|/k = {s_d}^2$. It outputs a uniformly random $s_d \times s_d$ square of blocks.

▶ **Theorem 6.** *Let $D_G \sim$ Greedy(H, k), $D_\square \sim$ Square(H, k). For $n_1 \cdot n_2 \cdots n_{d-2} \ge k \ge 2$, let $L = \arg\min\{\ell : n_1 \cdot n_2 \cdots n_\ell \ge k\}$.*

$$\mathbb{E}(\mathsf{Frag}(D_G)) \le \frac{k-1}{k^2}\sum_{\ell=1}^{L} n_\ell + \frac{1}{4}\sum_{\ell=L+1}^{d-1} n_\ell; \quad \mathbb{E}(\mathsf{Frag}(D_\square)) \ge \frac{2}{3}\left(\frac{\sqrt{n_1 \ldots n_{d-1}}}{\sqrt{k}} - \frac{11}{2}\right)\sqrt{n_{d-1}}.$$

Dallas County is nearly a perfect square shape, so it gives us an opportunity to set some roughly realistic parameters to evaluate these bounds. There are 529 tracts in Dallas County, with an average of 3.2 blocks groups per tract and 26.4 blocks per block group, yielding 44,113 total blocks. We can approximate these parameters by setting $d = 4$, using $k = 4$ as for the county commission districts, and setting $(n_1, n_2, n_3) = (484, 4, 25)$ which has a reasonably similar 48,400 blocks (as a result, $L = 1$). The bounds in the theorem say that $\mathbb{E}(\mathsf{Frag}(D_G)) \leq 98$ and $\mathbb{E}(\mathsf{Frag}(D_\square)) \geq 264$. Note: for homogeneous hierarchies H with equal-population leaves, the score $\mathsf{Frag}(D_G)$ is independent of algorithm randomness and can be computed exactly; for the above parameters $\mathsf{Frag}(D_G) = 90.75$. So the bound in the theorem is fairly tight, at least in this case.

To interpret the theorem, it is helpful to think of Greedy as being hierarchically greedy and Square as being geometrically greedy. That is, the former is oriented toward using the biggest possible units and keeping them whole, so that spatial considerations are secondary; the latter is oriented towards "compact" geographies with a lot of area relative to perimeter, and unit integrity is secondary. The theorem shows that compactness alone (a function of the plane geometry) does not keep down the fragmentation score (a function of the hierarchy), and indeed the bounds get farther apart as the hierarchy gets larger and more complicated. In Appendix C, we compare these theoretical results to empirical district errors, finding that fragmentation tracks well with errors in ToyDown, but that the complexity of the TopDown model weakens the relationship, suggesting a need for more sophisticated tools.

7 Ecological regression with noise

7.1 Inference methods for Voting Rights Act enforcement

When elections are conducted by secret ballot, it is fundamentally impossible to precisely determine voting patterns by race from the reported outcomes alone. The standard methods for estimating these patterns use the cast votes at the precinct level, combined with the demographics by precinct, to infer racial polarization. Because the general aggregate-to-individual inference problem is called "ecological" (cf. ecological paradox, ecological fallacy), the leading techniques are called *ecological regression* (ER) and *ecological inference* (EI). It is rare that EI and ER do not substantively agree, and we focus on ER here because it lends itself to easily interpretable pictures.

ER is a simple linear regression, fitting a line to the data points determined by the precincts on a demographics-vs-votes plot. A high slope (positive or negative) indicates a likely strong difference in voting preferences, which is necessary to demonstrate the Gingles 2-3 tests for a VRA lawsuit.

The top row of Figure 5 shows standard ER run on the precincts of Dallas County, with each precinct plotted according to its percentage of Hispanic voting age population or HVAP (x-axis) and the share of cast votes that went to Lupe Valdez (y-axis). Strong racial polarization would show up as a fit line of high slope. This process produces a point estimate of Hispanic support for Valdez, found by intersecting the fit line with the $x = 1$ line, which represents the scenario of 100% Hispanic population. The point estimate of non-Hispanic support for Valdez is at the intersection of the fit line with $x = 0$.

7.2 Summary of Experiments

ToyDown and TopDown were both run on the full Texas reconstruction from 2010. We plotted Dallas County votes from three contests: votes for Obama for president in 2012 general election, votes for Valdez for governor in the 2018 Democratic Party primary runoff, and

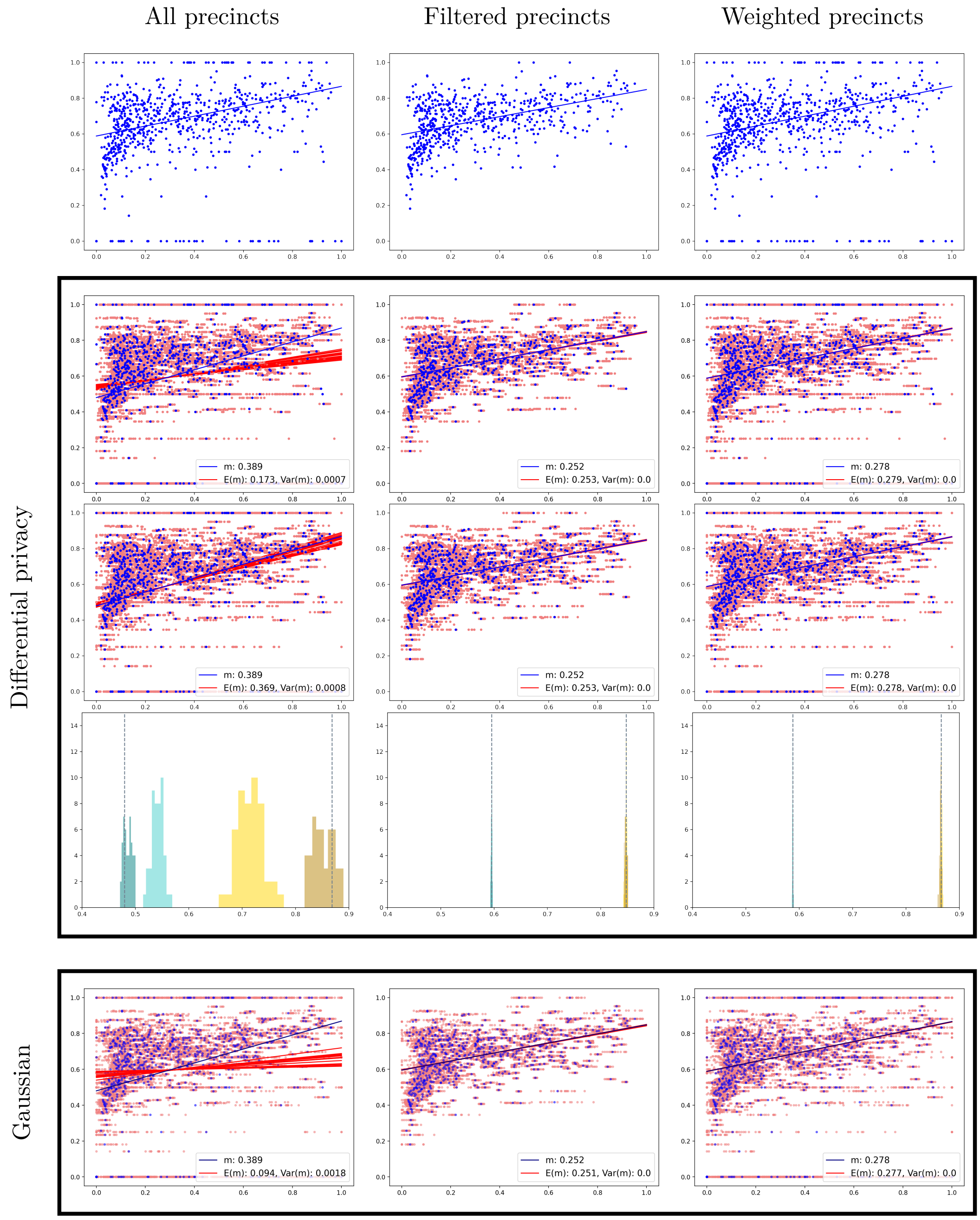


Figure 5 Comparing ecological regression on un-noised data (top row) with various styles of noising. ER is re-run on data noised by differentially private ToyDown (second row), and data noised by TopDown (third row), both with $\varepsilon = 1$, equal split. The blue dots repeat the un-noised data, the pink dots show 16 runs of noised data with pink fit lines re-computed each time. Below that, the histograms show the point estimates of Latino (gold) and non-Latino (teal) support for Valdez estimated from ER on data noised by ToyDown (lighter) and TopDown (darker). The last row contrasts the differentially private algorithms with a naive variant that adds noise to each precinct from a mean-zero Gaussian distribution, set to match the average precinct level L^1 error observed in the ToyDown runs (in this case, this is $\sigma = 20$). Across all of these experiments, the conclusion is striking: TopDown performs better than ToyDown and far better than a naive Gaussian variant, even without filtering precincts; if precincts are filtered or weighted, none of the noising alternatives threatens the ability to detect racially polarized voting.

Table 3 Point estimates from ER for Dallas County in the Valdez/White primary runoff in 2018. In the first table, estimates are made with (un-noised) VAP data from the 2010 Census. In the *filtered precincts* case, precincts with fewer than 10 cast votes are excluded from the initial set of 827 precincts. In the *weighted precincts* case, precincts are weighted by the number of cast votes. The ToyDown and TopDown estimates are made from VAP data from 16 runs with $\epsilon = 1$ and an ϵ-budget with all levels given equal weighting. Variance is the empirical variance over the repeated runs of the noising algorithm and is in units of 10^{-8}, shown to two significant digits.

	All precincts (827)		Filtered precincts (626)		Weighted precincts (827)	
Race	this group	complement	this group	complement	this group	complement
Hispanic	0.869	0.480	0.848	0.596	0.866	0.588
Black	0.917	0.518	0.851	0.620	0.835	0.595
White	0.555	0.623	0.474	0.811	0.478	0.805

			All (827)		Filtered (626)		Weighted (827)	
Race	Algorithm	statistic	group	compl.	group	compl.	group	compl.
Hispanic	ToyDown	mean	0.715	0.541	0.848	0.595	0.867	0.588
Hispanic	ToyDown	variance	36000	7000	250	43	160	19
Black	ToyDown	mean	0.798	0.543	0.851	0.62	0.835	0.595
Black	ToyDown	variance	39000	2100	89	5.9	25	2.1
White	ToyDown	mean	0.476	0.674	0.473	0.811	0.478	0.805
White	ToyDown	variance	17000	8000	64	36	33	17
Hispanic	TopDown	mean	0.853	0.485	0.848	0.595	0.865	0.587
Hispanic	TopDown	variance	45000	6700	480	100	120	16
Black	TopDown	mean	0.91	0.52	0.85	0.62	0.835	0.595
Black	TopDown	variance	30000	1200	250	23	45	2.4
White	TopDown	mean	0.582	0.607	0.472	0.81	0.47	0.804
White	TopDown	variance	10000	3400	92	37	92	10

votes for Chevalier for comptroller in the 2018 general election. We chose these contests because in each, ER finds evidence of strong racially polarized voting when using published 2010 census data. All three contests gave similar findings; we'll choose the Valdez runoff contest as our focus here.

For both ToyDown and TopDown, we vary how we handle the inclusion of small precincts in the ecological regression. The options are All (every precinct is a data point in the scatterplot, all weighted equally); Filtered (only including precincts with at least 10 votes cast in that election); or Weighted (weighting the terms in the objective function in least-squares fit by number of votes cast). Filtering and weighting are done using the exact number of cast votes, not the differentially private precinct population totals, which is realistic to the use case.

For each noising run we have a block- or precinct-level matrix, $\hat{M}$ of noised counts, with height b, the number of geographic units (blocks or precincts), and width c, the number of attributes for which there are counts recorded. We also have a corresponding matrix M of un-noised counts. We can compute the L_1 error by summing over the absolute value of every entry in $M - \hat{M}$. ToyDown and TopDown were run 16 times for each configuration. Let E_{avg} be the average L_1 error across noising runs.

If we add *Gaussian* noise to each count instead, the expected L_1 error is $\sum_{i,j} E[|X_{i,j}|]$, where $X_{i,j} \sim \mathcal{N}(0, \sigma^2)$. This is the half-normal distribution, so $E[|X_{i,j}|] = \frac{\sigma\sqrt{2}}{\sqrt{\pi}}$. We rearrange to find the standard deviation $\sigma = \frac{E_{avg}\sqrt{\pi}}{bc\sqrt{2}}$ that defines the Gaussian distribution (with $\mu = 0$), so that adding a random variable drawn from it to each unit count will produce an expected L^1 error matching the average E_{avg} observed across the runs.

7.3 The role of small precincts

Practitioners who use ER have raised two questions regarding the effect of differential privacy: (1) How robust will the estimate be after the noising? (2) Will noising diminish the estimate of candidate support from a minority population? We analyzed the effects of TopDown and ToyDown on the 2018 Texas Democratic primary runoff election, where Lupe Valdez was a clear minority candidate of choice in Dallas county.[1]

We begin by observing that of the 827 precincts in Dallas County, 201 have fewer than 10 cast votes from that election day – in fact, 99 precincts recorded zero cast votes. These precincts are a big driver of instability under DP. This is not surprising; percentage swings are much higher in small numbers even if the noise injected might be low. However, down-weighting these small precincts makes the estimate almost always agree with the un-noised estimate. Specifically, we assign weights to the precincts equivalent to the number of total votes in the precinct. Figure 5 shows how the estimates vary by run type and data treatment.

8 Conclusion

The central goal of this study has been to take the concerns of redistricting practitioners seriously and to investigate potential destabilizing effects of TopDown on the status quo. A second major goal is to make recommendations, both to the Disclosure Avoidance team at the Census Bureau and to the same practitioners – the attorneys, experts, and redistricting line-drawers in the field. Texas generally, and Dallas County in particular, was selected because it has been the site of several interesting Voting Rights Act cases in the last 20 years.[2]

Our top-line conclusion is that, at least for the Texas localities and election data we examined, TopDown performs far better than more naive noising in terms of preserving accuracy and signal detection for election administration and voting rights law. Perhaps more importantly, we have created an experimental apparatus to help other groups conduct independent analyses.

This work has led us to isolate several elements of common redistricting practice that lead to higher-variance outputs and more error under TopDown. The first example is the common use of a full precinct dataset, with no population weighting, in running racial polarization inference techniques. The second major example is the use of the smallest available units, census blocks, for building districts of all sizes, with no particular priority on intactness for larger units of Census geography. In both cases, we find that these were already likely sources of silent error. Filtering small precincts (or, better, weighting by population) and building districts that prioritize preserving whole the largest units that are suited to their scale are two examples of simple updates to redistricting practice. Besides being sound on first principles, these adjustments can insulate data users from DP-related distortions and help safeguard the important work of fair redistricting.

[1] We also examined the general elections for President in 2012 and Comptroller in 2018, with similar findings.

[2] This is a large county with considerable racial and ethnic diversity. Follow-up work will consider smaller and more racially homogeneous localities.

References

1 *13 U.S.C. Section 9*. URL: `https://www.law.cornell.edu/uscode/text/13/9`.

2 John Abowd, Daniel Kifer, Brett Moran, Robert Ashmead, Philip Leclerc, William Sexton, Simson Garfinkel, and Ashwin Machanavajjhala. Census topdown: Differentially private data, incremental schemas, and consistency with public knowledge, 2019. URL: `https://github.com/uscensusbureau/census2020-das-e2e/blob/master/doc/20190711_0945_Consistency_for_Large_Scale_Differentially_Private_Histograms.pdf`.

3 Avery v. Midland County, 390 U.S. 474 (1968).

4 U.S. Census Bureau. *Disclosure avoidance system - End to End demonstration*. URL: `https://github.com/uscensusbureau/census2020-das-e2e`.

5 U.S. Census Bureau. *Disclosure avoidance system - End to End demonstration, L1 metric*. URL: `https://github.com/uscensusbureau/census2020-das-e2e/blob/3f2c9cf9cb3c33a4e2067bd784ff381792f7ffc0/programs/validator.py#L20`.

6 U.S. Census Bureau. *TopDown: Adding Geometric Noise to Counts*. URL: `https://github.com/uscensusbureau/census2020-das-e2e/blob/d9faabf3de987b890a5079b914f5aba597215b14/programs/engine/topdown_engine.py#L678`.

7 U.S. Census Bureau. *2010 Demonstration Data Products*, 2010. URL: `https://www.census.gov/programs-surveys/decennial-census/2020-census/planning-management/2020-census-data-products/2010-demonstration-data-products.html`.

8 U.S. Census Bureau. *2010 Census Summary File 1*, 2012. URL: `https://www.census.gov/prod/cen2010/doc/sf1.pdf`.

9 U.S. Census Bureau. *Census P.L. 94-171 Redistricting Data*, 2017. URL: `https://www.census.gov/programs-surveys/decennial-census/about/rdo/summary-files.html`.

10 Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of markov chains for redistricting. *arXiv preprint arXiv:1911.05725*, 2019.

11 Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

12 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Halevi S., Rabin T. (eds) Theory of Cryptography. TCC 2006. Lecture Notes in Computer Science*, 3876, 2006.

13 Peter Wayner JN Matthews, Bhushan Suwal. *Accompanying GitHub repository*. URL: `https://github.com/mggg/census-diff-privacy`.

14 Denis Kazakov. *Census Scripts GitHub repository*, 2019. URL: `https://github.com/94kazakov/census_scripts`.

15 U.S. Census Bureau Michael Hawes. *Differential Privacy and the 2020 Decennial Census*, 2020. URL: `https://www2.census.gov/about/policies/2020-03-05-differential-privacy.pdf`.

16 National Conference of State Legislatures. *2010 Redistricting Deviation Table*. URL: `https://www.ncsl.org/research/redistricting/2010-ncsl-redistricting-deviation-table.aspx`.

17 Reynolds v. Sims, 377 U.S. 533 (1964).

18 Wesberry v. Sanders, 376 U.S. 1 (1964).

A ToyDown and TopDown

ToyDown is described in Algorithm 2. It uses the *Laplace distribution* $\text{Lap}(b)$ with scale parameter b, i.e., the probability distribution over $\mathbb{R}$ with mean zero and probability density function $\mathbb{P}[L] = \frac{1}{2b}e^{-|L|/b}$. It has variance $2b^2$. TopDown uses the *geometric* distribution, a discretized version of the Laplace distribution with integer support.

The inputs to TopDown are as follows. $A_{H,T} = \{a_{h,t}\}_{h \in H, t \in T}$, where $a_{h,t}$ is the number of people in h of type t; $W = (Q_1, \ldots, Q_{|W|})$ is a *workload* consisting of a collection of histograms Q; $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_d)$ is a hierarchical allocation of the privacy budget, with $\varepsilon_\ell > 0$

at each level; $B : W \to [0,1]$ with $\sum_{Q \in W} B(Q) = 1$ is a probability vector describing the relative privacy budget on each histogram in the workload; *invariants* V; and *structural inequalities* S. We write $\boldsymbol{a}_h = \{a_{h,t}\}_{t \in T}$ (and $\boldsymbol{\alpha}_h$ analogously). For a query q, we write $q(\boldsymbol{a}_h) = \sum_{t \in q} a_{h,t}$ (and $q(\boldsymbol{\alpha}_h)$ analogously).

In the first stage (lines 2-5), a geometric random variable is added to the raw counts a to produce noised counts $\hat{a}$. In the second stage (lines 6-8), the noised counts are adapted to the nearest integer values that meet a collection of equality and inequality conditions. These equalities and inequalities, over the real numbers, describe a convex polytope; therefore the post-processing can be thought of geometrically as a closest-point projection to the integer points in the convex body under L^2 distance.

The noising stages of both ToyDown and TopDown are ε-differentially private for $\varepsilon = \sum_{\ell=1}^{d} \varepsilon_\ell$. In ToyDown, this stage can be viewed as generating a single histogram at each level ℓ using budget ε_ℓ. Following the Census Bureau, we use bounded differential privacy, wherein the global sensitivity of histogram queries is 2. In TopDown, the budget at level ℓ is further divided among the $|W|$ histograms Q in the workload, each receiving $B(Q)\varepsilon_\ell$ of the budget. Because ToyDown's post-processing is data independent, ToyDown is ε-DP. TopDown's post-processing is not data independent: the invariants and structural inequalities may depend on the original data.

Algorithm 1 TopDown, based on [2].

```
1: **procedure** TopDown($A_{H,T}$, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d$, $W$, $B$, $V$, $S$)
2:     **for** $h \in H$, $Q \in W$, $q \in Q$ **do**
3:         $\beta \leftarrow \exp(-B(Q) \cdot \varepsilon_{\ell(h)}/2)$
4:         $G_{h,q} \leftarrow \mathrm{Geom}(\beta)$                          ▷ See [6]
5:         $\widehat{a}_{h,q} \leftarrow q(\boldsymbol{a}_h) + G_{h,q}$          ▷ Geometric mechanism with
                                                  sensitivity 2, budget $B(Q) \cdot \varepsilon_{\ell(h)}$
6:     **for** $\ell = 1, \ldots, d$ **do**
7:         Compute hierarchically-consistent              ▷ A sophisticated heuristic algorithm
           non-negative integers $\{\alpha_{h,t}\}_{h \in H_\ell, t \in T}$           out of scope for this work
           minimizing $\sum_{h \in H_\ell} \sum_{q \in W_\ell} (q(\boldsymbol{\alpha}_h) - \widehat{a}_{h,q})^2$,
           subject to the invariants: $v^*(\boldsymbol{\alpha}_h) = v^*(\boldsymbol{a}_h)$ for all $h \in H_\ell$, $v \in V$
           and structural inequalities: $s(\boldsymbol{\alpha}_h, \boldsymbol{a}_h) \le 0$ for all $h \in H_\ell$, $s \in S$
8:     **return** $\{\alpha_{h,t}\}_{h \in H, t \in T}$
```

B Detailed materials and methods

B.1 Primary data sources

2010 US Census demographic data was downloaded using the Census API, and the 2010 census block, block group, and tract shapefile for Dallas County were downloaded from the US Census Bureau's TIGER/Line Shapefiles. For our VRA analysis, we obtained both statewide election results and a statewide precinct shapefile from the Texas Capitol Data Portal, which we then trimmed to the precincts within Dallas County.[3]

[3] Data comes from `data.capitol.texas.gov/topic/elections` and `data.capitol.texas.gov/topic/geography`.

Algorithm 2 ToyDown.

```
procedure ToyDown($A_H = \{a_h\}_{h \in H}, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d$)        ▷ (Single attribute)
    for $h \in H$ do
        $L_h \sim \mathrm{Lap}(2/\varepsilon_{\ell(h)})$
        $\widehat{a}_h \leftarrow a_h + L_h$        ▷ Laplace mechanism with sensitivity 2, budget $\varepsilon_{\ell(h)}$
    for $\ell = 1, \ldots, d$ do
        Compute hierarchically consistent $\{\alpha_h\}_{h \in H_\ell}$
        minimizing $\sum_{h \in H_\ell} (\alpha_h - \widehat{a}_h)^2$
    return $\{\alpha_h\}_{h \in H}$

procedure MultiAttrToyDown($A_{H,T} = \{a_{h,t}\}_{h \in H, t \in T}, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d$)
    for $h \in H$, $t \in T$ do
        $L_{h,t} \sim \mathrm{Lap}(2/\varepsilon_{\ell(h)})$
        $\widehat{a}_{h,t} \leftarrow a_{h,t} + L_{h,t}$        ▷ Laplace mechanism with sensitivity 2, budget $\varepsilon_{\ell(h)}$
    for $\ell = 1, \ldots, d$ do
        Compute hierarchically consistent
        (optionally, non-negative) $\{\alpha_{h,t}\}_{h \in H_\ell, t \in T}$
        minimizing $\sum_{h \in H_\ell, t \in T} \left(\alpha_{h,t} - \widehat{a}_{h,t}\right)^2$
    return $\{\alpha_{h,t}\}_{h \in H, t \in T}$
```

We use a person-level dataset obtained by applying a reconstruction technique to the block-level data from Texas from the 2010 Census.[4] The reconstructed microdata records contain block-level sex, age, ethnicity, and race information consistent with a collection of tables from 2010 Census Summary File 1. We note that this reconstruction follows the same strategy used by the Census Bureau itself as the first step of its reidentification experiment [15], based on [11].

The reconstructed data is far from perfect. Unlike the Bureau, we do not have access to the ground truth data needed to quantify the errors. The Bureau's own reconstruction experiment reconstructed 46% of entries exactly, plus an additional 25% within ±1 year error in age [15]. We note that our reconstructed data contains no household information, because this was not present in the tables used in the constraint system. This is significant because the TopDown configurations for the US Census Bureau's 2010 Demonstration Data Products [7] include household-based workload queries and invariants.

B.2 TopDown configuration

The exact configuration files and code for all the runs are available in this paper's accompanying repository [13]. The TopDown code used for this paper was modified from the publicly available demonstration release of the US Census Bureau's Disclosure Avoidance System 2018 End-to-End test release [4]. The input data fed to the algorithm was obtained by restructuring the reconstructed 2010 block-level Texas microdata into the 1940s IPUMs data format. Most importantly, the reconstructions allowed for 63 distinct combination of races whereas the End-to-End release only allows for 6 races, so all multi-racial entries were re-categorized as Other in our TopDown runs.

[4] A team led by data scientist and journalist Mark Hansen at Columbia, including Denis Kazakov, Timothy Donald Jones, and William Reed Palmer, designed an algorithm to solve for the detailed data, which we describe in this section. Code is available upon request [14].

Because TopDown's post-processing is done level by level, the noisy counts in Dallas County do not depend on the noisy counts at the tract-level or below in counties other than Dallas. We modified the census reconstructed data to focus on Dallas county and minimize the computation time spent processing the other 253 counties in Texas. Specifically, for every non-Dallas county, we placed all of the population into a single block.

We do not enforce certain household invariants that the Census Bureau is planning to enforce, and our workload omits household queries that are used in Census's demonstration data products. Our choice to omit household queries and invariants is result of our use of reconstructed 2010 census microdata which does not include household information. We did perform additional runs with household invariants and queries using crude synthetic household data, the results of which are available in the data repository accompanying this paper [13]. In those runs, the population in each block was grouped into households of size 5 with at most one group smaller than 5. Ultimately, we focused on the experiments that did not require synthetic household data.

The TopDown runs without the household workload or invariants use a workload consisting of two histograms: $Q_{detailed}$ and $Q_{va,eth,race}$ with 10% and 90% of the budget respectively. (The additional runs with households includes an additional households and group quarters histogram in the workload assigned 22.5% of the budget, leaving 10% and 67.5% for $Q_{detailed}$ and $Q_{va,eth,race}$ respectively.) The End-to-End TopDown code reports a differentially private estimate of the L^1 error with $\varepsilon = 0.0001$ not included in privacy budget specified elsewhere in the configuration file and discussed elsewhere in this paper.

C District fragmentation

Algorithm 3 Greedy.

```
procedure GREEDY($H, k$)
    if $k = 1$ then
        Return $H$
    $N \leftarrow \lfloor |H_d|/k \rfloor$, $D \leftarrow \emptyset$, $h^* \leftarrow h_1$
    while $N > 0$ do
        For $h^*$ and $D$, let $S(h^*, D)$ be the set of
        children $h$ of $h^*$ that are disjoint from $D$.
        while $\exists h \in S(h^*, D) : |h| \leq N$ do
            $D \leftarrow D \cup h$                ▷ Associating $h$ with the blocks descendent from it
            $N \leftarrow N - |h|$
        Pick $h^* \in S(h^*, D)$
    return $D$
```

Algorithm 4 Square.

```
procedure SQUARE($H, k$)
    $s_d \leftarrow \sqrt{|H_d|/k}$                 ▷ Side length in blocks of the district
    $S_d \leftarrow \sqrt{n_1 \cdot n_2 \cdots n_{d-1}}$       ▷ Side length in blocks of the region
    Sample $i, j \in \{1, \ldots, S_d - s_d + 1\}$ uniformly at random
    return $D_{i,j}$, the square district with top left corner at $(i, j)$
```

In Section 6, we defined the fragmentation score and its relationship to error variance for ToyDown, and analyzed the expected fragmentation score of districts produced by different district drawing algorithms. Now we apply TopDown to examine the relationship between a district's population error and geometry, as captured by the fragmentation score.

We fix the a total budget and an equal allocation across levels: $0.2 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon_5 = \varepsilon_6$, as in Table 1. (We do not need to noise the nation because we are focusing on Texas; we do need to noise Texas even though its total population is invariant, because its population by race is allowed to vary.) We apply ReCom to build districts out of tracts, block groups, and blocks – all of which are part of the census hierarchy – and add a realistic variant that builds from whole *precincts*. These are about the same size as block groups and are more commonly used in redistricting.

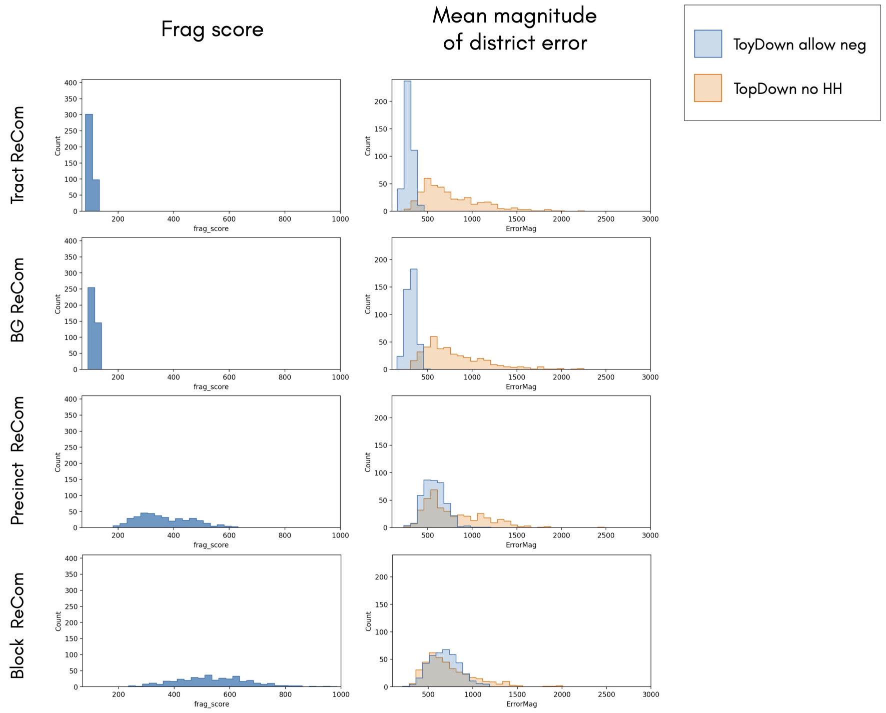


Figure 6 Do the building-block units of districts matter? Histograms of fragmentation score (left column) and mean error magnitude (right column) are shown across four district-drawing algorithms that prioritize compactness. (Dallas County, $k = 4$.) We see that using larger units leads to significantly lower fragmentation and correspondingly low district-level error in ToyDown, but the advantage erodes when we pass to TopDown.

Figure 6 plots the data from our experiments. Each of the 12 histograms displays 400 values, one for each district drawn by the specified district-drawing algorithm. The histograms on the left plot the fragmentation score of each district; the histograms on the right plot the mean observed district-level population error magnitude over 16 executions of the specified hierarchical noising algorithm.

The size of the constituent units is observed to have a controlling effect on the fragmentation score, as expected. As we would expect, this carries over to the simplest ToyDown (allowing negativity). (Note that since the error has zero mean, higher variance drives up the mean magnitude of error.) But the choice of base units makes far less difference by the time we move to full TopDown. These observations are consistent, again, with a strong similarity across spatially nearby units. All four kinds of ReCom will tend to produce compact, squat districts whose units are more closely geographically proximal than would be observed with disconnected or elongated shapes. Random noise is uncorrelated, but the post-processing effects can be highly spatially correlated because of spatial relationships in the underlying counts by race, ethnicity, and voting age.

D Robustness of noisy ER

Figure 7 extends the findings from Figure 5 with more splits and allocations, showing that as long as small precincts are filtered out, ecological regression for RPV analysis in Dallas County is robust to changes in the allocation of the privacy budget across the levels of the hierarchy and the total privacy budget for TopDown. The corresponding plots for ToyDown are essentially indistinguishable. (ER with precincts weighted by population is similarly robust.)

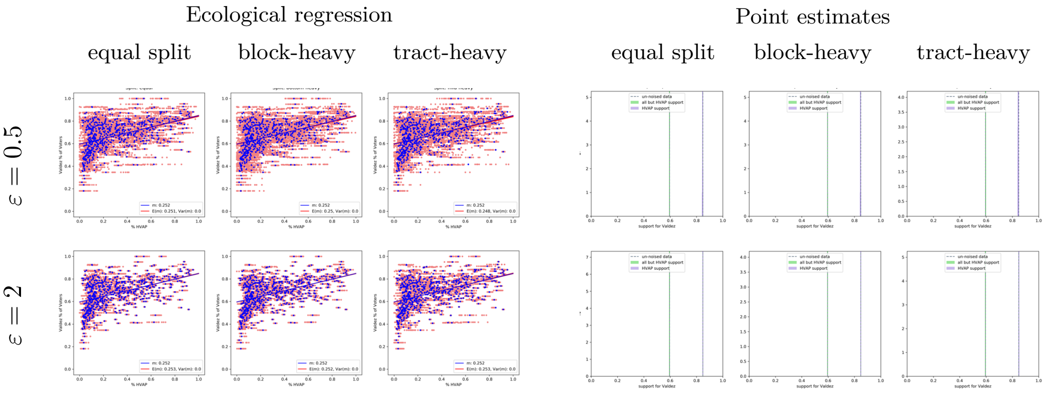


Figure 7 Ecological regression for the Valdez-White runoff election with $\varepsilon = .5$ and $\varepsilon = 2$ and three different budget allocations, together with corresponding point estimates for Latino and non-Latino support for Valdez, with small precincts filtered out as in Figure 5. Findings stay remarkably stable.

Lexicographically Fair Learning: Algorithms and Generalization

Emily Diana ✉
University of Pennsylvania, Philadelphia, PA, USA

Wesley Gill ✉
University of Pennsylvania, Philadelphia, PA, USA

Ira Globus-Harris ✉
University of Pennsylvania, Philadelphia, PA, USA

Michael Kearns ✉
University of Pennsylvania, Philadelphia, PA, USA

Aaron Roth ✉
University of Pennsylvania, Philadelphia, PA, USA

Saeed Sharifi-Malvajerdi ✉
University of Pennsylvania, Philadelphia, PA, USA

Abstract

We extend the notion of minimax fairness in supervised learning problems to its natural conclusion: *lexicographic* minimax fairness (or *lexifairness* for short). Informally, given a collection of demographic groups of interest, minimax fairness asks that the error of the group with the *highest* error be minimized. Lexifairness goes further and asks that amongst all minimax fair solutions, the error of the group with the second highest error should be minimized, and amongst all of *those* solutions, the error of the group with the third highest error should be minimized, and so on. Despite its naturalness, correctly defining lexifairness is considerably more subtle than minimax fairness, because of inherent sensitivity to approximation error. We give a notion of approximate lexifairness that avoids this issue, and then derive oracle-efficient algorithms for finding approximately lexifair solutions in a very general setting. When the underlying empirical risk minimization problem absent fairness constraints is convex (as it is, for example, with linear and logistic regression), our algorithms are provably efficient even in the worst case. Finally, we show generalization bounds – approximate lexifairness on the training sample implies approximate lexifairness on the true distribution with high probability. Our ability to prove generalization bounds depends on our choosing definitions that avoid the instability of naive definitions.

2012 ACM Subject Classification Computing methodologies → Machine learning

Keywords and phrases Fair Learning, Lexicographic Fairness, Online Learning, Game Theory

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.6

Related Version *Full Version*: https://arxiv.org/abs/2102.08454

Funding Supported in part by the Warren Center for Network and Data Sciences, NSF grant CCF-1763307 and the Simons Collaboration on the Theory of Algorithmic Fairness.

1 Introduction

Most notions of statistical group fairness ask that a model approximately equalize some error statistic across demographic groups. Often this is motivated as a tradeoff: the goal is to lower the error of the most disadvantaged group, and if doing so requires increasing the error on some more advantaged group, so be it – this is a cost that we are willing to pay in the name of equity. But solutions which equalize group errors do *not* in general mediate a clean tradeoff in which losses in accuracy on more advantaged groups result in increases in

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 6; pp. 6:1–6:23
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

accuracy on less advantaged groups: instead, generically (i.e. except in the very special case in which the Bayes optimal error is identical for all groups), a constraint of equalizing group error rates may require *artificially increasing* the error on at least one group, without any corresponding benefit to any other group.

A partial answer to this criticism of standard notions of group fairness is the classical notion of *minimax fairness*, recently studied by [23, 9] in the context of supervised learning. Minimax fairness asks for a model which minimizes the error of the group *most disadvantaged* by the model – i.e. the group with maximum group error. In doing so, it realizes the promise of equal error solutions in that it trades off higher error on populations more advantaged by the model for lower error on populations less advantaged by the model when this is possible – but without artificially increasing the error of any group when doing so. Indeed, it is not hard to see that a minimax model necessarily *weakly Pareto dominates* an equal error rate model, in the sense that group errors are only lower in the minimax solution *simultaneously for all groups.*

This narrative is most sensible if there are only two demographic groups of interest. If there are more than two groups, there may be many different minimax optimal models that have very different error profiles for groups other than the max error group. How should we choose amongst these? Prior work [9] has broken ties by optimizing for overall classification accuracy. But why should we entirely give up on the goal of optimizing for the most disadvantaged, partially enunciated in the motivation of minimax fairness, once we have fixed the error of only one of many groups?

In this paper we propose the natural continuation of this idea, which we call *lexicographic minimax fairness.* Informally speaking, this notion recurses on the idea that we wish to minimize the cost of the least well off. A model that satisfies lexicographic fairness, which we call a *lexifair* model, will minimize the maximum error γ_1 on any group, amongst all possible models (i.e. a lexifair model is a also a minimax model). Further, amongst the set of all minimax models, a lexifair model must minimize the error of the group with the second highest error γ_2. Amongst all of these models, it further minimizes the error of the group with the third highest error γ_3, and so on.[1]

1.1 Our Contributions

Our first contribution is a definition of (approximate) lexicographic minimax fairness. Correctly defining an actionable notion of lexicographic minimax fairness is surprisingly subtle. For standard computational and statistical reasons, it will not be possible to exactly match the distributional lexicographically optimal error rates $\gamma_1, \gamma_2, \gamma_3$, etc. But as we will observe, these lexicographically optimal error rates can be arbitrarily unstable, in the sense that amongst the set of models that have minimax error larger than γ_1 by even an arbitrarily small margin, the value of the optimal lexifair error on the third highest error group γ_3' can be arbitrarily larger than γ_3 (See our example in Section 2.1.1). An implication of this is that the vectors of errors γ, γ' representing exact lexifair solutions in and out of sample can be entirely incomparable and arbitrarily different from one another. Hence we need a definition of approximate lexifairness that accounts for this instability, and allows for sensible statements about approximation and generalization.

Another challenge arises in the interaction between our definitions and our (desired) algorithms. A constraint on the *highest* error amongst all groups, which arises in defining minimax error, is convex, and hence amenable to algorithmic optimization. However, naive

[1] It is easy to see that there are cases in which a lexifair model may have arbitrarily smaller errors than a minimax model on all but the worst-off group.

specifications of lexifairness involve constraints on the second highest group errors, the third highest group errors, and more generally kth highest errors. These are non-convex constraints when taken in isolation. However, as it turns out, a constraint on the second highest error becomes convex when we restrict attention to minimax optimal classifiers, and more generally, a constraint on the kth highest error becomes convex once the values of the lower order group errors are constrained to their lexifair values. We show this by giving a clearly convex variant of our lexifair definition, specified by exponentially many *linear constraints*, which replace constraints on the k'th highest error groups with constraints on the *sums* of all k-tuples of group errors. We then show that our definition of "convex lexifairness" is equivalent to our original notion of lexifairness, at least in the exact case (absent approximation). We give our formal definitions in Section 2.1.2.

With our notion of approximate lexifairness in hand and our convexified constraints, we give *oracle-efficient* algorithms for finding approximate lexifair models in both the regression and classification case. This means that our algorithms are efficient reductions to the problem of unconstrained (that is, standard non-fair) learning over the same model class. Despite the worst-case intractability of most natural learning problems even absent fairness considerations, a desirable feature of oracle-efficient algorithms is that they can be implemented using any of the common and practical heursitics for non-fair learning, often with good empirical success [20, 30, 16, 1].

Our algorithms are based on solving the corresponding constrained optimization problem by recasting it as a (Lagrangian) minmax optimization problem, and using no-regret dynamics. Because our "convexified" lexifairness constraints are exponentially numerous, the "constraint player" in our formulation has exponentially many strategies – but as we show, we can efficiently optimize over her strategy space using an efficient separation oracle. Hence the constraint player can always play according to a "best response" strategy in our simulated dynamics. When our base model class is continuous and our loss function convex (as it is with e.g. linear regression), then the "learner" in our dynamics can play gradient descent over parameter space. In this case, our oracle efficient-algorithms are in fact fully polynomial time algorithms because our reduction to weighted learning problems involves only *non-negative* weights, which preserves convexity. In the classification case, when our loss function is *non-convex*, we can convexify it by considering the set of all probability distributions over base models. Here the parameters we optimize over become the weights of the probability distribution, and our loss function (i.e. the expected loss over the choice of a random model) becomes linear in our (enormous) parameter space. In this case, we are effectively solving a linear program that has both exponentially many variables and exponentially many constraints – but we are nevertheless able to do so in an oracle-efficient manner by making appropriate use of the Follow the Perturbed Leader algorithm [18] for no-regret learning.

Finally, we prove a generalization theorem, showing that if we have a dataset S (sampled i.i.d. from an underlying distribution) that has sufficiently many samples from each group, and if we have a model that is approximately lexifair for S, then the model is also approximately lexifair on the underlying distribution. This is significantly more involved than just a standard uniform convergence argument – which would simply state that our in and out of sample errors on each group are close to one another – because approximate lexifairness additionally depends on the precise *relationship* between these group errors. Nevertheless, we show that uniform convergence is a sufficient condition to guarantee that in-sample lexifairness bounds correspond to out of sample lexifairness bounds.

1.2 Related Work

There are many notions of group or statistical fairness that are studied in the fair machine learning literature, which are generally concerned with *equalizing* various measures of error across protected groups; see e.g. [4, 24] for surveys of many such metrics.

Minimax solutions are a classical approach to fairness that have been used in many contexts including scheduling, fair division, and clustering (see e.g. [14, 3, 29, 5, 6]). A number of these works employ techniques for solving two-player zero-sum games as part of their algorithmic solution [6, 5]. This is the same general algorithmic framework that we use. More recently, minimax group error has been proposed as a fairness solution concept for classification problems in machine learning [23, 9, 22]. These works generally do not specify how to choose between multiple minimax solutions, with the exception of [9], which gives algorithms for choosing the solution with smallest overall classification error subject to the minimax constraint.

Lexicographic minimax fairness has been studied in the fair division literature for tasks such as quota allocation in mobile networks, load balancing, and network design [10, 7, 25, 33, 32, 28, 2, 27, 26]. As far as we know, we are the first to study lexicographic fairness in a learning context in which the quantities of interest must be *estimated*, and hence the first to identify the sensitivity issues that arise when defining *approximate* notions of lexicographic fairness.

An alternative approach to learning one classifier for all groups is to learn *decoupled classifiers* [11, 31], i.e. a separate classifier for each group. The decoupling of error rates across all groups eliminates tradeoffs between groups, and hence results in classifiers that are lexicographically fair (within the class of decoupled classifiers). But there are at least three important reasons one might want to learn a single classifier (the approach we take) rather than a separate classifier for each group. The first is that learning separate classifiers for each group requires that the groups be *disjoint*, which is not needed in our approach. For example, we could divide the population into groups according to race, gender, and age – despite the fact that individuals will fall into multiple groups simultaneously. In other words, our algorithms can be used to obtain *subgroup* or *intersectional* fairness [19, 20, 15, 21, 17, 13]. Second, learning separate classifiers for each group requires that protected group membership be used explicitly at classification time, which can be undesirable or illegal in important applications. Finally, learning a single classifier allows for the possibility of transfer learning, whereby a small sample from some group can be partially made up for by larger quantities of data from other (nevertheless related) groups.

2 Model and Definitions

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be an arbitrary data domain. Each data point in our setting is a pair $z = (x, y)$ where $x \in \mathcal{X}$ is the feature vector and $y \in \mathcal{Y}$ is the response variable (i.e. the label). Let $\mathcal{X}$ consist of points belonging to K (not necessarily disjoint) groups $\mathcal{G}_1, \ldots, \mathcal{G}_K$, so we can write $\mathcal{X} = \cup_{k=1}^{K} \mathcal{G}_k$. We write $\mathcal{P}$ to denote an arbitrary distribution over $\mathcal{Z}$, and $\mathcal{P}_k$ to denote the marginal distribution induced by $\mathcal{P}$ on the kth group $\mathcal{G}_k \times \mathcal{Y}$. Let $S = \{z_i\}_{i=1}^{n}$ be a data set of size n, which for the purposes of proving generalization bounds, we will take to consist of n data points drawn i.i.d. from $\mathcal{P}$. Denote the points in S that are contained in $\mathcal{G}_k$ by $G_k \times \mathcal{Y}$, so we can write $S = \cup_{k=1}^{K} G_k$.

Let $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{Y}\}$ be the model class of interest, and let $L : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ be a loss function that takes a data point z and a model h as inputs, and outputs the loss of h on z. For instance, in the case of classification and zero-one loss, we have $L(h, z) = \mathbb{1}\left[h(x) \neq y\right]$. We will abuse notation and write $L_z(\cdot)$ for $L(\cdot, z)$ for any data point z. Throughout the paper, for any distribution $\mathcal{P}$, we write the expected loss of a model h over $\mathcal{P}$ as:

$$L_{\mathcal{P}}(h) \triangleq L(h, \mathcal{P}) \triangleq \mathbb{E}_{z \sim \mathcal{P}}\left[L_z(h)\right].$$

We slightly abuse notation and write $L_S(h)$ to denote the empirical loss on a dataset S. Here and throughout the paper when S plays the role of a distribution, we interpret that as the *uniform distribution* over the points in S, and accordingly, $z \sim S$ as a point sampled *uniformly at random* from S.

Until Section 7, we will work exclusively with sample quantities, and so for simplicity of notation, let us define $L_k(h) \triangleq L_{G_k}(h)$ to denote the *sample* loss of a model h on the k'th group. When necessary, we will write $L_k\left(h, \mathcal{P}\right)$ to denote $L_{\mathcal{P}_k}\left(h\right)$, the corresponding *distributional* loss of h on the k'th group. For any model h and any data set $S = \cup_k \{G_k\}$, let $\bar{h}_S$ be the ordering induced on the groups $\{G_k\}_{k=1}^K$ by the loss of h, breaking ties arbitrarily. In other words, $\bar{h}_S : [K] \to [K]$ is any bijection such that the following condition holds: $L_{\bar{h}_S(1)}(h) \geq L_{\bar{h}_S(2)}(h) \geq \ldots \geq L_{\bar{h}_S(K)}(h)$. The corresponding distributional ordering of the groups by any model h is defined similarly: for any model h and any distribution $\mathcal{P}$ over $\mathcal{Z}$, let $\bar{h}_{\mathcal{P}} : [K] \to [K]$ be the ordering induced on the groups $\{\mathcal{G}_k\}_{k=1}^K$ by the expected loss of h, breaking ties arbitrarily. In other words, $\bar{h}_{\mathcal{P}}$ is any bijection such that the following condition holds: $L_{\bar{h}_{\mathcal{P}}(1)}(h, \mathcal{P}) \geq L_{\bar{h}_{\mathcal{P}}(2)}(h, \mathcal{P}) \geq \ldots \geq L_{\bar{h}_{\mathcal{P}}(K)}(h, \mathcal{P})$. When the distribution (data set) is clear from context, we elide the dependence on the distribution (data set) and simply write $\bar{h}$ for $\bar{h}_{\mathcal{P}}$ ($\bar{h}_S$).

Our definition of lexifairness will be given recursively. At the base level, we define $\mathcal{H}_{(0)} = \mathcal{H}$ to be the set of all models in our class. Then recursively for all $1 \leq j \leq K$, we define:

$$\gamma_j \triangleq \min_{h \in \mathcal{H}_{(j-1)}} L_{\bar{h}(j)}(h), \quad \mathcal{H}_{(j)} \triangleq \left\{ h \in \mathcal{H}_{(j-1)} : L_{\bar{h}(j)}(h) = \gamma_j \right\}.$$

In words, γ_j is the smallest error that any model in $\mathcal{H}_{(j-1)}$ obtains on the group that has the jth highest error, and $\mathcal{H}_{(j)}$ is the set of *all* models in $\mathcal{H}_{(j-1)}$ that attain this minimum – i.e. that have jth highest error equal to γ_j. Thus, γ_1 is the minimax error – i.e. the highest group error for the model that is chosen to *minimize* the maximum group error. Similarly, γ_2 is the error of the second highest group for all minimax optimal models that further minimize the error of the second highest group, and so on. With this notation in hand, we can define exact lexifairness as follows:

▶ **Definition 1** (Exact Lexicographic Fairness). *Let $1 \leq \ell \leq K$. We say a model $h \in \mathcal{H}$ satisfies level-ℓ (exact) lexicographic fairness (lexifairness) if for all $j \leq \ell$, $L_{\bar{h}(j)}(h) \leq \gamma_j$.*

Minimax fairness corresponds to level-1 lexifairness. This is a definition of *exact* lexifairness, in that it permits no approximation to the error rates – i.e. we require $L_{\bar{h}(j)}(h) \leq \gamma_j$ for all j, and hence $L_{\bar{h}(j)}(h) = \gamma_j$ for all j. For a variety of reasons, we will need definitions that tolerate approximation. For example, because we inevitably have to train on a fixed dataset, but want our guarantees to generalize to new datasets drawn from the same distribution, we will need to accommodate statistical approximation. The optimization techniques we will bring to bear will also only be able to *approximate* lexifairness, even in sample. But it turns out that defining a sensible approximate notion of lexifairness is more subtle than it first appears.

2.1 Approximate Lexifairness: Stability and Convexity

We begin with the "obvious" but ultimately flawed definition of approximate lexifairness (Definition 2), and then explain why it is lacking in stability. This will lead us to the definitions we finally adopt: Definition 3 and its *convexified* version (Definition 5), which we show is equivalent (Claim 7), and for which we can develop efficient algorithms.

2.1.1 The Challenge of Stability

The most natural seeming definition of approximate lexifairness begins with our notion of exact lexifairness (Definition 1), and adds slack to all of the inequalities contained within. In other words, we attempt to find a model that has sorted group errors $\gamma'_1, \gamma'_2, \ldots, \gamma'_K$ that pointwise approximate the optimal lexifair vector of sorted group errors $\gamma_1, \ldots, \gamma_K$.

▶ **Definition 2** (A Flawed Definition). *Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-lexicographic fairness if for all $j \leq \ell$, $L_{\bar{h}(j)}(h) \leq \gamma_j + \alpha$.*

To see the problem with the above definition, consider a setting with three groups, and a model class $\mathcal{H}$ that contains all distributions (or randomized classifiers) over two pure classifiers $\{h_1, h_2\}$. Imagine that h_1 induces the (unsorted) vector of group error rates $\langle 0.5, 0.5, 0 \rangle$, and h_2 induces the (unsorted) vector of group error rates $\langle 0.5 + 2\alpha, 0, 0.5 \rangle$, for some arbitrarily small $\alpha > 0$. Note that it is easy to construct distributions over labeled instances with exactly these group error vectors by simply arranging each classifier to disagree with the labels on the specified fraction of a group. So, for simplicity we abstract away the data and directly discuss the error vectors.

The minimax group error for this model class is $\gamma_1 = 0.5$, and is achieved only by h_1 which has error 0.5 on the first and second groups. Since the largest group error of h_2 is also on the first group with value $0.5 + 2\alpha > 0.5$, any distribution over $\{h_1, h_2\}$ that places a non-zero probability on h_2 will therefore violate the (exact) minimax constraint. This in turn implies that $\mathcal{H}_{(1)} = \{h_1\}$. Therefore, the only exact lexifair model is h_1 and thus $\gamma_1 = 0.5$, $\gamma_2 = 0.5$, $\gamma_3 = 0$.

However, imagine that because of estimation error (as is inevitable if we are learning based on a finite sample) or optimization error (since we generally don't have access to exact optimization oracles in learning settings), we slightly misestimate the minimax group error γ_1 to be $\gamma'_1 = 0.5 + \alpha$. If we now optimize, allowing the largest group error to be as much as $\gamma'_1 = 0.5 + \alpha$, we may now find randomized classifiers which put weight as large as 0.5 on h_2. The uniform distribution over $\{h_1, h_2\}$ induces the unsorted vector of group errors $\langle 0.5 + \alpha, 0.25, 0.25 \rangle$. The induced error on the second group (which is now also the group with second largest error) of 0.25 is considerably *smaller* than $\gamma_2 = 0.5$. So far this appears to be all right, since $\gamma'_2 < \gamma_2$. But if we now attempt to optimize the error of the third highest error γ'_3, *subject to the constraint* that the largest group error is (close to) γ'_1 and the second largest group error is (close to) γ'_2, we now find that we are forced to settle for third highest group error $\gamma'_3 \approx 0.25$, which is considerably *larger* than the value of the third highest group's error of $\gamma_3 = 0$ in the exact lexifair solution.

This example highlights a fundamental *instability* of our first (flawed) attempt at defining approximate lexifairness: even arbitrarily small estimation (or optimization) error introduced to the minimax error rate γ_1 can result in large, non-monotonic effects for later group errors – enforcing even a valid *upper bound* on γ_1 can cause γ_3 to increase substantially, and these effects compound even further if we have more than three groups.

2.1.2 A Stable and Convex Definition

With the proceeding example of the instability inherent in our (flawed) Definition 2, we now give the definition of approximate lexifairness that we begin with:

▶ **Definition 3** (Approximate Lexicographic Fairness). *Fix a distribution $\mathcal{P}$. Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. For any sequence of mappings $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, define $\mathcal{H}^{\vec{\epsilon}}_{(0)}(\mathcal{P}) \triangleq \mathcal{H}$, and recursively for all $1 \leq j \leq \ell$ define:*

$$\mathcal{H}^{\vec{\epsilon}}_{(j)}(\mathcal{P}) \triangleq \left\{ h \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P}) : L_{\bar{h}(j)}(h, \mathcal{P}) \leq \min_{g \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} L_{\bar{g}(j)}(g, \mathcal{P}) + \epsilon_j(h) \right\}$$

and let $\|\vec{\epsilon}\|_\infty = \max_{1 \leq j \leq \ell} \max_{h \in \mathcal{H}} \epsilon_j(h)$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-lexicographic fairness ("lexifairness") with respect to $\mathcal{P}$ if there exists $\vec{\epsilon}$ with $\|\vec{\epsilon}\|_\infty \leq \alpha$ such that for all $j \leq \ell$:

$$L_{\bar{h}(j)}(h, \mathcal{P}) \leq \min_{g \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} L_{\bar{g}(j)}(g, \mathcal{P}) + \epsilon_j(h) + \alpha.$$

When we prove bounds on empirical lexifairness, we simply take the distribution to be the uniform distribution over the data set S. When the distribution is clear from context, we will write $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ and elide the dependence on the distribution.

Note that there are two distinctions between Definition 3 and Definition 2. First, the recursively defined sets $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ now incorporate some $\epsilon_j(\cdot)$ slack in their parameterization which will help capture statistical (or optimization) error. Second (and crucially), we now call a solution (ℓ, α)-approximately lexifair if it satisfies our requirements for *some* sequence of relaxations $\vec{\epsilon}$ that is component-wise less than α for all models h. It is this second point that avoids the instability and non-monotonicity that arises from Definition 2. We observe that Definition 3 is a strict weakening of Definition 2:

▶ **Claim 4.** *Definition 3 is a relaxation of Definition 2: if a model satisfies (ℓ, α)-lexicographic fairness according to Definition 2, then it also satisfies (ℓ, α)-lexicographic fairness according to Definition 3.*

Proof. If a model satisfies (ℓ, α)-lexicographic fairness according to Definition 2, then by taking $\vec{\epsilon} = \vec{0}$, it also meets the conditions of Definition 3. ◀

We now face another definitional challenge. A priori, Definition 3 appears to be highly non-convex, because it constrains the second highest group error, the the third highest group error, etc.[2] This is in contrast to standard equal-error notions of fairness, or minimax fairness (which constrains only the highest group error) that *are* convex in the sense that a distribution over fair models remains fair. Without convexity of this sort, the algorithmic problem of finding a fair model becomes much more challenging. But in fact (at least for $\alpha = 0$), Definition 3 *does* give a convex constraint. To see this, we first introduce an alternative notion of *convex lexifairness*, and then show that it actually represents the exact same constraint as lexifairness when the approximation parameter $\alpha = 0$.

[2] E.g., if we have two groups and two models which induce group errors $(0.5, 0)$ and $(0, 0.5)$ respectively, both solutions have a second-highest error of 0 – but convex combinations have a second highest error strictly greater than 0. So absent other structure, upper bounding the second highest group error of a model corresponds to a non-convex constraint. But note that in this two-group example, the non-convexity dissapears if we restrict attention to minimax optimal models. This is what we will take advantage of more generally.

▶ **Definition 5** (Convex Lexicographic Fairness). *Fix a distribution $\mathcal{P}$. Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. For any sequence of mappings $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, define $\mathcal{F}^{\vec{\epsilon}}_{(0)}(\mathcal{P}) \triangleq \mathcal{H}$, and recursively for all $1 \leq j \leq \ell$ define:*

$$\mathcal{F}^{\vec{\epsilon}}_{(j)}(\mathcal{P}) \triangleq$$
$$\left\{ h \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P}) : \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \epsilon_j(h) \right\}$$

and let $\|\vec{\epsilon}\|_\infty = \max_{1 \leq j \leq \ell} \max_{h \in \mathcal{H}} \epsilon_j(h)$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-convex lexicographic fairness with respect to $\mathcal{P}$ if there exists $\vec{\epsilon}$ with $\|\vec{\epsilon}\|_\infty \leq \alpha$ such that for all $j \leq \ell$:

$$\max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \epsilon_j(h) + \alpha.$$

When we prove bounds on empirical convex lexifairness, we simply take the distribution to be the uniform distribution over the data set S. When the distribution is clear from context, we will write $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ and elide the dependence on the distribution.

Here, we have replaced constraints on the j'th highest group error with constraints on the *sum* of group errors over all $\approx K^j$ subsets of groups of size j. This has replaced a single constraint with many constraints, but each is convex, and hence the resulting set of constraints defined by $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ is convex. We will formally prove this in the following claim.

▶ **Claim 6** (Convexity of $\mathcal{F}^{\vec{\epsilon}}_{(j)}$). *Let $L_z : \mathcal{H} \to \mathbb{R}_{\geq 0}$ be a convex loss function. If the initial model class $\mathcal{H}$ is convex, then for all j and all $\vec{\epsilon}$ such that the mappings $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$ are concave, the set $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ is convex.*

The proof can be found in the full version of the paper ([8]), and proceeds by straightforward induction. We note that while some classes of models naturally satisfy the convexity conditions of the above claim with respect to their corresponding parameters (e.g. linear and logistic regression), this claim will apply to arbitrary classification models with zero-one loss as well. In these settings, we will convexify the class of models by considering the set of all probability distributions over deterministic models. The loss of a distribution (i.e. a randomized model) is then defined as the *expected* loss, when the model is sampled from the corresponding distribution. Hence, by linearity of expectation, our loss functions will be convex (linear) in the parameters – i.e. the weights – of these distributions.

It turns out that our notion of *convex* lexifairness is identical to our notion of lexifairness (and so our original definition in fact specified a convex set of constraints), at least when the approximation parameter $\alpha = 0$. We prove this in the following claim:

▶ **Claim 7** (Relationship between $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ and $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ when $\vec{\epsilon} = \vec{0}$). *For all j, and $\vec{\epsilon} = \vec{0}$, we have $\mathcal{F}^{\vec{\epsilon}}_{(j)} = \mathcal{H}^{\vec{\epsilon}}_{(j)}$.*

The intuition for the claim is the following. The sets $\mathcal{H}_{(j)}$ in Definition 3 constrain the error of the group that has the j'th highest error. In contrast, the sets $\mathcal{F}_{(j)}$ from Definition 5 constrain the *sum* of the errors for all possible j-tuples of groups. Amongst all of these constraints, the binding one will be the constraint corresponding to the j groups that have the *largest* errors. But because (inductively) the errors of the top $j-1$ error groups have already been appropriately constrained in $\mathcal{F}_{(j-1)}$, this reduces to a constraint on the j'th highest error group, as desired. These constraints are numerous, but each is convex, and so the resulting set of constraints can be seen to be convex. See the full version of the paper ([8]) for the formal proof of Claim 7, which proceeds by induction.

We emphasize that despite the complexity of our final Definition 5, what we have shown is that it is in fact a relaxation of our initial, natural definition of exact lexifairness (Definition 1) – and in particular Definitions 1, 3, and 5 coincide exactly when $\alpha = 0$. We do not know the precise relationship between our definitions of approximate lexifairness and approximate convex lexifairness for $\alpha > 0$ – but because both are smooth relaxations of the same base definition, both should be viewed as capturing the same intuition as Definition 1 (exact lexifairness) when α is small.

3 Game Theory and No-Regret Learning Preliminaries

3.1 No-Regret Dynamics

In this subsection, we briefly review the seminal result of Freund and Schapire [12]: Under certain conditions, two-player zero-sum games can be (approximately) solved by having access to a no-regret online learning algorithm for one of the players.

Suppose in this subsection that S_1 and S_2 are two vector spaces over the field of real numbers. Consider a zero-sum game with two players: a player with strategies in S_1 (the minimization player) and another player with strategies in S_2 (the maximization player). Let $U : S_1 \times S_2 \to \mathbb{R}_{\geq 0}$ be the payoff function of this game. For every strategy $s_1 \in S_1$ of player one and every strategy $s_2 \in S_2$ of player two, the first player gets utility $-U(s_1, s_2)$ and the second player gets utility $U(s_1, s_2)$.

▶ **Definition 8** (Approximate Equilibrium). *A pair of strategies $(s_1, s_2) \in S_1 \times S_2$ is said to be a ν-approximate minimax equilibrium of the game if the following conditions hold:*

$$U(s_1, s_2) - \min_{s_1' \in S_1} U(s_1', s_2) \leq \nu, \qquad \max_{s_2' \in S_2} U(s_1, s_2') - U(s_1, s_2) \leq \nu$$

In other words, (s_1, s_2) is a ν-approximate equilibrium of the game if neither player can gain more than ν by deviating from their strategies.

Freund and Schapire [12] proposed an efficient framework for approximately solving the game: In an iterative fashion, have one of the players play according to a no-regret learning algorithm, and let the second player (approximately) best respond to the play of the first player. The empirical average of each player's actions over a sufficiently long sequence of such play will form an approximate equilibrium of the game. The formal statement is given in the following theorem.

▶ **Theorem 9** (No-Regret Dynamics [12]). *Let S_1 and S_2 be convex, and suppose the utility function U is convex-concave: $U(\cdot, s_2) : S_1 \to \mathbb{R}_{\geq 0}$ is convex for all $s_2 \in S_2$, and $U(s_1, \cdot) : S_2 \to \mathbb{R}_{\geq 0}$ is concave for all $s_1 \in S_1$. Let $(s_1^1, s_1^2, \ldots, s_1^T)$ be the sequence of play for the first player, and let $(s_2^1, s_2^2, \ldots, s_2^T)$ be the sequence of play for the second player. Suppose for $\nu_1, \nu_2 \geq 0$, the regret of the players jointly satisfies*

$$\sum_{t=1}^{T} U(s_1^t, s_2^t) - \min_{s_1 \in S_1} \sum_{t=1}^{T} U(s_1, s_2^t) \leq \nu_1 T, \qquad \max_{s_2 \in S_2} \sum_{t=1}^{T} U(s_1^t, s_2) - \sum_{t=1}^{T} U(s_1^t, s_2^t) \leq \nu_2 T$$

Let $\bar{s}_1 = \frac{1}{T} \sum_{t=1}^{T} s_1^t \in S_1$ and $\bar{s}_2 = \frac{1}{T} \sum_{t=1}^{T} s_2^t \in S_2$ be the empirical average play of the players. We have that the pair $(\bar{s}_1, \bar{s}_2)$ is a $(\nu_1 + \nu_2)$-approximate equilibrium of the game.

No regret online learning algorithms are algorithms that can guarantee the conditions of Theorem 9 against arbitrary adversaries. We will use two no-regret online learning algorithms: *Online Projected Gradient Descent*, which we will use in regression settings in which models are represented by parameters in a Euclidean space, and *Follow the Perturbed Leader (FTPL)*, which we will use in binary classification settings ([8]). We will make use of these no-regret learning algorithms in our proposed algorithm for learning a lexifair model; full explanations and pseudocode are in Appendix C.

4 Finding Lexifair Models

In this section we focus on developing the tools required to prove the following (informally stated) theorem. Formal claims appear in Theorems 13 (regression) and 14 (classification).

▶ **Theorem 10** (Informal). *Suppose the model class $\mathcal{H}$ is convex and compact, and that the loss function $L_z : \mathcal{H} \to \mathbb{R}_{\geq 0}$ is convex for all data points $z \in \mathcal{Z}$. There exists an efficient algorithm that returns a model which is (ℓ, α)-convex lexicographic fair (according to Definition 5), for any given ℓ and α.*

We will propose algorithms for both classification and regression settings. The algorithms we propose proceed inductively to solve the minimax problems defined recursively by our convex lexifair definition. The first minimax problem is the one that minimizes the maximum group error rate: $\min_{h\in\mathcal{H}} \max_{k\in[K]} L_k(h)$. Let us denote the estimated value (computed by the first phase of our algorithm) for this minimax problem by η_1. The second minimax problem is minimizing the maximum sum of any two group error rates subject to the constraint that all group error rates are at most η_1: the estimated value for this minimax problem is called η_2. The rest of the minimax problems are defined in a similar inductive fashion: suppose at round $j \leq \ell$, we are given some estimates $(\eta_1, \ldots, \eta_{j-1})$ for the first $j-1$ minimax values. Now using these estimates, the new minimax problem for the sum of any j group error rates can be stated as follows.

$$\min_{\substack{h\in\mathcal{H}: \\ \forall r\leq j-1,\, \forall\{i_1,\ldots,i_r\}\subseteq[K] \\ L_{i_1}(h)+\ldots+L_{i_r}(h)\leq\eta_r}} \left\{ \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h) \right\}. \tag{1}$$

We can reformulate this problem by calling the objective $\max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h) := \eta_j$ and introducing a new set of constraints which require that any sum of j group error rates must be at most η_j. Note that this new formulation introduces a new variable, η_j, to the optimization problem. We therefore have that the optimization problem (1) is equivalent to

$$\min_{\substack{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]: \\ \forall r\leq j,\, \forall\{i_1,\ldots,i_r\}\subseteq[K] \\ L_{i_1}(h)+\ldots+L_{i_r}(h)\leq\eta_r}} \eta_j \triangleq \mathrm{OPT}_j\left(\eta_1,\ldots,\eta_{j-1}\right) \tag{2}$$

which is a constrained convex optimization problem given that the model class $\mathcal{H}$ and the loss function L are convex. Here $L_M = \max_{z,h} L_z(h)$ is an upper bound on the loss function which identifies the range of feasible values for η_j: $[0, j\cdot L_M]$. Recall that in this round, $(\eta_1, \ldots, \eta_{j-1})$ are given from the previous rounds, and η_j is a variable in the optimization problem. We denote the optimal value of the optimization problem (2) by $\mathrm{OPT}_j\left(\eta_1,\ldots,\eta_{j-1}\right)$.

4.1 Formulation as a Two-Player Zero-Sum Game

Optimization problem (2) is written as a constrained optimization problem, but we can express it equally well as an unconstrained minimax problem via Lagrangian duality. The corresponding Lagrangian can be written as:

$$\mathcal{L}_j\left((h,\eta_j),\lambda\right) = \eta_j + \sum_{r=1}^{j} \sum_{\{i_1,\ldots,i_r\}\subseteq[K]} \lambda_{\{i_1,i_2,\ldots,i_r\}} \cdot \left(L_{i_1}(h) + \ldots + L_{i_r}(h) - \eta_r\right) \tag{3}$$

where we introduce one dual variable λ for every inequality constraint in the optimization problem (2), and index the dual variables by their corresponding constraint. Therefore, there are $q_j = \sum_{r=1}^{j}\binom{K}{r}$ dual variables in this round. Solving optimization problem (2) is equivalent to solving the following minimax problem:

$$\min_{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]} \max_{\lambda\in\mathbb{R}^{q_j}_{\geq 0}} \mathcal{L}_j\left((h,\eta_j),\lambda\right) = \max_{\lambda\in\mathbb{R}^{q_j}_{\geq 0}} \min_{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]} \mathcal{L}_j\left((h,\eta_j),\lambda\right) \tag{4}$$

where the minimax theorem holds because 1) the range of the primal variables, i.e. $\mathcal{H}$ and $[0, j \cdot L_M]$, is convex and compact, the range for the dual variable ($\mathbb{R}^q_{\geq 0}$) is convex, and 2) $\mathcal{L}_j\left((h,\eta_j),\lambda\right)$ is convex in its primal variables (h,η_j) and concave in the dual variable λ. Therefore we focus on solving the minimax problem (4) which can be seen as solving a two-player zero-sum game with payoff function $\mathcal{L}_j\left((h,\eta_j),\lambda\right)$. Using the no-regret dynamics of [12] (see Section 3.1), we will have the primal player (or *Learner*) with strategies $(h,\eta_j) \in \mathcal{H} \times [0, j \cdot L_M]$ play a no-regret learning algorithm and let the dual player (or *Auditor*) with strategies $\lambda \in \Lambda_j = \{\lambda \in \mathbb{R}^{q_j}_{\geq 0} : \|\lambda\|_1 \leq B\}$ best respond. Here we place an upper bound B on the ℓ_1-norm of the dual variable to guarantee convergence of our algorithms. This nuisance parameter will be set optimally in our algorithms, and we note that the minimax theorem continues to hold in the presence of this upper bound on λ. We will first analyze the best response problem for both players – i.e. the problem of optimizing the Lagrangian for one of the players *fixing* the strategy of the other player.

4.2 The Auditor's Best Response

Fixing the (h,η_j) variables of the Learner and the estimated values $(\eta_1,\ldots,\eta_{j-1})$ from previous rounds, the Auditor can best respond by solving

$$\operatorname*{argmax}_{\lambda\in\Lambda_j} \mathcal{L}_j\left((h,\eta_j),\lambda\right) \equiv \operatorname*{argmax}_{\lambda\in\Lambda_j} \sum_{r=1}^{j} \sum_{\{i_1,\ldots,i_r\}\subseteq[K]} \lambda_{\{i_1,i_2,\ldots,i_r\}} \cdot \left(L_{i_1}(h) + \ldots + L_{i_r}(h) - \eta_r\right).$$

Since the objective is linear in the dual variables λ, the Auditor can without loss of generality best respond by putting all its mass B on the variable $\lambda_{\{i_1,i_2,\ldots,i_r\}}$ corresponding to the most violated constraint, if one exists. In particular, given any model $h \in \mathcal{H}$ and any ordering $\bar{h}$ induced by h on the groups, we have that the Auditor's best response $\lambda_{\text{best}}(h,\eta_j)$ is

$$\lambda_{\text{best}}(h,\eta_j) = \begin{cases} 0 \in \mathbb{R}^{q_j} & \text{if } \forall r \leq j :\, L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) \leq \eta_r \\ \lambda^\star \in \mathbb{R}^{q_j} & \text{if } \exists r \leq j :\, L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) > \eta_r \end{cases}$$

where the entries of $\lambda^\star$ are defined as follows.

$$\lambda^\star_{\{i_1,i_2,\ldots,i_r\}} = \begin{cases} B & \text{if } \{i_1,i_2,\ldots,i_r\} = \{\bar{h}(1),\bar{h}(2),\ldots,\bar{h}(r^\star)\} \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

where $r^\star \in \operatorname{argmax}_{r\leq j}\left(L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) - \eta_r\right)$.

Note that the Auditor's best response can be computed efficiently because it only requires sorting the vector of error rates across K groups. We summarize the best response algorithm for the Auditor in Algorithm 1.

Algorithm 1 The Auditor's Best Response (λ_{best}): jth round.

```
Input: Learner's play $(h, \eta_j)$, previous estimates $(\eta_1, \dots, \eta_{j-1})$
Compute $L_k(h)$ for all groups $k \in [K]$;
Find the top $j$ elements of vector $(L_1(h), \dots, L_K(h))$ and call them:
 $L_{\bar{h}(1)}(h) \geq \dots \geq L_{\bar{h}(j)}(h)$;
if $\forall r \leq j : L_{\bar{h}(1)}(h) + \dots + L_{\bar{h}(r)}(h) \leq \eta_r$ then $\lambda_{out} = 0$;
else Let $r^\star \in \text{argmax}_{r \leq j} \left( L_{\bar{h}(1)}(h) + \dots + L_{\bar{h}(r)}(h) - \eta_r \right)$, $\lambda_{out} = \lambda^\star$ as in
 Equation (5) ;
Output: $\lambda_{out} \in \Lambda_j$
```

4.3 The Learner's Best Response

Given dual weights $\lambda \in \Lambda_j$ chosen by the Auditor, the Learner can best respond by solving

$$\underset{h \in \mathcal{H}, \eta_j \in [0, j \cdot L_M]}{\text{argmin}} \quad \mathcal{L}_j \left((h, \eta_j), \lambda \right).$$

We note that the objective function $\mathcal{L}_j \left((h, \eta_j), \lambda \right)$ can be decomposed into three terms: one that depends only on the model h, another that depends only on η_j, and finally one that is constant (with respect to (h, η_j)). Therefore, this optimization problem is separable for the Learner – the decomposition is formally described below.

$$\mathcal{L}_j \left((h, \eta_j), \lambda \right) = \mathcal{L}_j^1 \left(h, \lambda \right) + \mathcal{L}_j^2 \left(\eta_j, \lambda \right) + C_j \left(\lambda \right) \tag{6}$$

where

$$\mathcal{L}_j^1 \left(h, \lambda \right) \triangleq \sum_{r=1}^{K} w_r(\lambda) L_r(h), \text{ where } w_r(\lambda) \triangleq \sum_{s=0}^{j-1} \sum_{\{i_2, \dots, i_s\} \subseteq [K] \setminus \{r\}} \lambda_{\{r, i_2, \dots, i_s\}} \tag{7}$$

$$\mathcal{L}_j^2 \left(\eta_j, \lambda \right) \triangleq \left(1 - \sum_{\{i_1, \dots, i_j\} \subseteq [K]} \lambda_{\{i_1, i_2, \dots, i_j\}} \right) \eta_j \tag{8}$$

$$C_j \left(\lambda \right) \triangleq - \sum_{r=1}^{j-1} \sum_{\{i_1, \dots, i_r\} \subseteq [K]} \lambda_{\{i_1, i_2, \dots, i_r\}} \cdot \eta_r \tag{9}$$

Given this decomposition of the Lagrangian, the best response (h, η_j) of the Learner to the variables λ of the Auditor is as follows:

$$(h, \eta_j) = \left(\underset{h \in \mathcal{H}}{\text{argmin}} \, \mathcal{L}_j^1 \left(h, \lambda \right), \underset{\eta_j \in [0, j \cdot L_M]}{\text{argmin}} \mathcal{L}_j^2 \left(\eta_j, \lambda \right) \right).$$

Note that the first optimization problem is a weighted minimization problem over the class $\mathcal{H}$, and the second one is a simple minimization of a linear function. Furthermore, even though in general computing the sums in Equations (7) and (8) can be computationally hard (because they are sums over exponentially many terms), *when the Auditor is best responding (which will be the case in our algorithms), these sums can be computed efficiently.* We formally state this claim in Fact 11.

▶ **Fact 11.** *When the Auditor is using its best response algorithm (Algorithm 1) to respond to the Learner, the Auditor will either output zero or identify a single subset C of groups ($|C| \leq j$) on which the constraints are violated maximally. In the former case, $w_r(\lambda) = 0$ for all r and $1 - \sum_{\{i_1,\dots,i_j\}\subseteq[K]} \lambda_{\{i_1,i_2,\dots,i_j\}} = 1$. In the latter case, we have*

$$w_r(\lambda) = B \cdot \mathbb{1}\left[r \in C\right], \quad 1 - \sum_{\{i_1,\dots,i_j\}\subseteq[K]} \lambda_{\{i_1,i_2,\dots,i_j\}} = 1 - B \cdot \mathbb{1}\left[|C| = j\right].$$

4.4 Solving the Game with No-Regret Dynamics

Having analyzed the best response problem for both players, we now focus on developing efficient algorithms to approximately solve the two-player zero-sum game defined above, which corresponds to finding an approximate convex lexifair model. The algorithms we propose use no-regret dynamics (see Section 3.1) in which the Learner plays a no-regret learning algorithm and the Auditor best responds according to Algorithm 1. As a consequence, we get that the empirical average of the played strategies $((\hat{h}, \hat{\eta}_j), \hat{\lambda})$ of the players over the course of the iterative algorithms will form a ν-approximate equilibrium of the game for some small value of $\nu \geq 0$ (according to Definition 8). Then, by the following theorem, we can turn these equilibrium guarantees into the fairness guarantees of the output model $\hat{h}$. Its proof can be found in [8].

We remark that what we mean by the empirical average will depend on the setting. If we are in a setting in which the loss function is convex in the model parameters (e.g. logistic or linear regression), then we can actually average the model parameters, and output a single deterministic model. Alternately, if we are in a classification setting in which the loss function (e.g. zero-one loss) is non-convex in the model parameters, then by averaging, we mean using the randomized model that corresponds to the uniform distribution over the empirical play history.

▶ **Theorem 12.** *At round j, let $(\hat{\eta}_1, \dots, \hat{\eta}_{j-1})$ be any given estimated minimax values from the previous rounds and let the strategies $((\hat{h}, \hat{\eta}_j), \hat{\lambda})$ form a ν-approximate equilibrium of the game for this round, i.e.,*

$$\mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \hat{\lambda}\right) \leq \min_{h\in\mathcal{H}, \eta_j\in[0, j\cdot L_M]} \mathcal{L}_j\left((h, \eta_j), \hat{\lambda}\right) + \nu, \quad \mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \hat{\lambda}\right) \geq \max_{\lambda\in\Lambda_j} \mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \lambda\right) - \nu.$$

We have that $\hat{\eta}_j \leq OPT_j\left(\hat{\eta}_1, \dots, \hat{\eta}_{j-1}\right) + 2\nu$, and for all $r \leq j$,

$$\max_{\{i_1,\dots,i_r\}\subseteq[K]} \sum_{s=1}^{r} L_{i_r}(\hat{h}) \leq \hat{\eta}_r + \frac{jL_M + 2\nu}{B}.$$

We will next instantiate this general result to give concrete algorithms for learning convex lexifair models in the regression and classification settings respectively.

5 Finding Lexifair Regression Models

Suppose in this section that $\mathcal{Y} \subseteq \mathbb{R}$ and $\mathcal{H}$ is a class of models in which each model is parametrized by some d-dimensional vector in $\mathbb{R}^d$: $\mathcal{H} = \{h_\theta : \theta \in \Theta\}$ where $\Theta \subseteq \mathbb{R}^d$. In this parametric setting we can think of each parameter $\theta \in \Theta$ as a model and write the loss function as a function of θ. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is differentiable for all z.[3] We will have the Learner play according to the Online Projected Gradient Descent

[3] If it is not differentiable we can use sub-gradients instead of gradients.

Algorithm 2 `LexiFairReg`: Finding a Lexifair Regression Model.

```
Input: $S = \cup_{k=1}^{K} G_k$ data set consisting of $K$ groups, $(\ell, \alpha)$ desired fairness
       parameters, loss function parameters $L_M$ and $G$, diameter $D$ of the model
       class $\Theta$
for $j = 1, 2, \ldots, \ell$ do
    Set $T_j = \frac{4j^2(GD+L_M)^2(2\alpha+jL_M)^2}{\alpha^4}$;
    Set $B_j = \frac{\alpha+jL_M}{\alpha}$;
    $(\hat{\theta}_j, \hat{\eta}_j) = \texttt{RegNR}(T_j, B_j; \hat{\eta}_1, \ldots, \hat{\eta}_{j-1})$ (Calling Algorithm 3)
end
Output: $(\ell, \alpha)$-convex lexifair model $\hat{\theta}_\ell$
```

algorithm (see Appendix C.1) where the gradients of the corresponding loss function of the game for the Learner (i.e. $\mathcal{L}_j\left((\theta, \eta_j), \lambda\right)$) can be computed using Equations (7) and (8), and the decomposition given in (6):

$$\nabla_\theta \mathcal{L}_j\left((\theta, \eta_j), \lambda\right) = \nabla_\theta \mathcal{L}_j^1\left(\theta, \lambda\right) = \sum_{r=1}^{K} w_r(\lambda) \nabla_\theta L_r(\theta), \tag{10}$$

$$\nabla_{\eta_j} \mathcal{L}_j\left((\theta, \eta_j), \lambda\right) = \nabla_{\eta_j} \mathcal{L}_j^2\left(\eta_j, \lambda\right) = 1 - \sum_{\{i_1, \ldots, i_j\} \subseteq [K]} \lambda_{\{i_1, i_2, \ldots, i_j\}}. \tag{11}$$

The algorithm for this setting is given as Algorithm 2, which makes calls to a subroutine (Algorithm 3) that solves the two-player zero-sum games defined above by having the Learner play Online Projected Gradient Descent (see Appendix C) and the Auditor best respond using Algorithm 1. Note that since the Auditor is best responding, computing the sums in Equations (10) and (11) can be done efficiently per Fact 11.

▶ **Theorem 13** (Lexifairness for Regression). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact, and bounded with diameter D: $\sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is convex and that there exists constants L_M and G such that $L_z(\cdot) \leq L_M$ and $\|\nabla_\theta L_z(\cdot)\|_2 \leq G$, for all data points $z \in \mathcal{Z}$. We have that for any $\ell \leq K$ and any $\alpha \geq 0$, the model $\hat{\theta}_\ell \in \Theta$ output by Algorithm 2 is (ℓ, α)-convex lexicographic fair.*

The proof of this theorem (which can be found in Appendix A) involves bounding the regret of each player, and then appealing to Theorem 12.

6 Finding Lexifair Classification Models

In this section we briefly discuss how we can find lexifair models in a classification setting. All details including our algorithm for this setting and its analysis can be found in [8]. Suppose in this section that $\mathcal{Y} = \{0, 1\}$ and our model class $\mathcal{H}$ is the probability simplex over a class of deterministic binary classifiers. We slightly abuse notation and write $\mathcal{H}$ for the given class of deterministic classifiers and write $\Delta\mathcal{H} \triangleq \{p : p \text{ is a distribution over } \mathcal{H}\}$ for the probability simplex, and work with $\Delta\mathcal{H}$ as our model class. Let the loss function be zero-one loss: for any $h \in \mathcal{H}$: $L_z(h) = \mathbb{1}\left\{h(x) \neq y\right\}$. The loss of any randomized model p on data point z is defined as the *expected loss* of h on z when h is sampled from $\mathcal{H}$ according to the distribution p. In other words,

$$L_z(p) \triangleq \mathbb{E}_{h \sim p}\left[L_z(h)\right]$$

Algorithm 3 RegNR: jth round.

```
Input: Number of rounds $T$, dual variable upper bound $B$, previous estimates
       $(\eta_1, \ldots, \eta_{j-1})$
Set learning rates $\eta = \frac{D}{jBG\sqrt{T}}$ and $\eta' = \frac{jL_M}{(1+B)\sqrt{T}}$;
Initialize the Learner: $\theta^1 \in \Theta, \eta_j^1 \in [0, j \cdot L_M]$;
for $t = 1, 2, \ldots, T$ do
    Learner plays $(\theta^t, \eta_j^t)$;
    Auditor best responds: $\lambda^t = \lambda_{\text{best}}(\theta^t, \eta_j^t; (\eta_1, \ldots, \eta_{j-1}))$ using Algorithm 1;
    Learner updates its actions using Projected Gradient Descent:
        $\theta^{t+1} = \text{Proj}_\Theta\left(\theta^t - \eta \cdot \nabla_\theta \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$
        $\eta_j^{t+1} = \text{Proj}_{[0, j \cdot L_M]}\left(\eta_j^t - \eta' \cdot \nabla_{\eta_j} \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$
    where the gradients are given in Equations (10) and (11).
end
Output: the average play $\hat{\theta} = \frac{1}{T}\sum_{t=1}^T \theta^t \in \Theta$, and $\hat{\eta}_j = \frac{1}{T}\sum_{t=1}^T \eta_j^t \in [0, j \cdot L_M]$.
```

which is convex (linear) in the model p (weights of the distribution). We will also assume that the model class $\mathcal{H}$ has finite VC dimension. Sauer's Lemma will then imply that for any finite dataset, $\mathcal{H}$ induces only finitely many labelings. This will serve two purposes. First, it allows us to write the optimization problem as a linear program with *finitely many* variables (probability weights over the set of all possible induced labelings), and therefore appeal to strong duality. Second, it allows us to pose the Learner's best response problem as an n-dimensional *linear optimization* problem, over the only exponentially many labelings of the n data points. This is what will allow us to apply Follow the Perturbed Leader and obtain oracle-efficient no-regret learning guarantees for the Learner. Here we are following an approach similar to that of [19]. The final algorithm will then have the Learner play according to Follow the Perturbed Leader (given access to a *Cost Sensitive Classification Oracle* for the function class $\mathcal{H}$), and have the Auditor best respond.

▶ **Theorem 14** (Lexifairness for Classification). *Let $\mathcal{H}$ be any class of binary classifiers with finite VC dimension, and let $L_z(p) = \mathbb{E}_{h\sim p}[L_z(h)]$ for any randomized model $p \in \Delta\mathcal{H}$ where $L_z(h) = \mathbb{1}\{h(x) \neq y\}$ is the zero-one loss. Fix any $\ell \leq K$ and any $\alpha \geq 0$. There exists an oracle-efficient algorithm (see [8]) such that for any $\delta > 0$, with probability at least $1-\delta$, its output model is (ℓ, α)-convex lexicographic fair.*

7 Generalization

In this section, we turn our attention to out of sample bounds. Standard uniform convergence statements would tell us that if we have enough samples from every group, then our in-sample group errors are good estimates of our out of sample group errors. However, this alone does not directly imply that we satisfy approximate lexifairness out of sample. We prove this is the case below. Our ability to prove out of sample bounds crucially relies on our definitional choices that removed the instability of the naive Definition 2. Specifically, we show that if:

1. Our base class $\mathcal{H}$ satisfies a standard uniform convergence bound across every group (so that we can control the maximum gap between in and out of sample error across every $h \in \mathcal{H}$, within each group k), and

2. We have a model that is approximately convex lexifair on our dataset $S \sim \mathcal{P}^n$, then then our model is also appropriately convex lexifair on the underlying distribution (with some loss in the approximation parameter).

▶ **Theorem 15** (Generalization for Convex Lexifairness). *Fix any distribution $\mathcal{P}$. Suppose for every $\delta > 0$, there exists $\beta(\delta)$ such that the following uniform convergence bound holds.*

$$\Pr_S \left[\max_{h \in \mathcal{H}, k \in [K]} \left| L_k\left(h, S\right) - L_k\left(h, \mathcal{P}\right) \right| > \beta(\delta) \right] < \delta$$

where S is a data set sampled i.i.d. from $\mathcal{P}$. We have that for every data set S sampled i.i.d. from $\mathcal{P}$, if a model h satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1 - \delta$ it also satisfies (ℓ, α')-convex lexicographic fairness with respect to $\mathcal{P}$ for $\alpha' = \alpha + 2\ell\beta(\delta)$.

The proof of the theorem is given in Appendix B. We can now instantiate the above theorem in a classification setting in which we have VC-type convergence bounds. A corollary that we get by applying standard uniform convergence bounds for finite VC classes is the following:

▶ **Corollary 16** (Generalization for Convex Lexifairness: Classification Setting). *Suppose $\mathcal{H}$ is a class of binary classifiers with VC dimension $d_{\mathcal{H}}$ and let $L_z(p) = \mathbb{E}_{h \sim p}\left[L_z(h)\right]$ for any randomized model $p \in \Delta\mathcal{H}$ where $L_z(h) = \mathbb{1}\left\{h(x) \neq y\right\}$ is the zero-one loss. We have that for every $\mathcal{P}$, every data set $S \equiv \{G_k\}_k$ of size n sampled i.i.d. from $\mathcal{P}$, if a model $p \in \Delta\mathcal{H}$ satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1 - \delta$ it also satisfies $(\ell, 2\alpha)$-convex lexicographic fairness with respect to $\mathcal{P}$ provided that*

$$\min_{1 \le k \le K} |G_k| = \Omega\left(\frac{l^2\left(d_{\mathcal{H}} \log\left(n\right) + \log\left(K/\delta\right)\right)}{\alpha^2}\right).$$

We have here proven a generalization theorem for convex lexifairness (Definition 5) which is the definition that our algorithms satisfy. We also prove a generalization theorem for lexifairness (Definition 3) which can be found in [8].

References

1 Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

2 M. Allalouf and Y. Shavitt. Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation. *IEEE/ACM Transactions on Networking*, 16(5):1015–1024, 2008. doi:10.1109/TNET.2007.905605.

3 Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010.

4 Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.

5 Robert S. Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4705–4714. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf.

6 Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 300–332, Chicago, Illinois, 22–24 March 2019. PMLR. URL: `http://proceedings.mlr.press/v98/cotter19a.html`.

7 Emilie Danna, Avinatan Hassidim, Haim Kaplan, Alok Kumar, Yishay Mansour, Danny Raz, and Michal Segalov. Upward max-min fairness. *J. ACM*, 64(1), 2017. `doi:10.1145/3011282`.

8 Emily Diana, Wesley Gill, Ira Globus-Harris, Michael Kearns, Aaron Roth, and Saeed Sharifi-Malvajerdi. Lexicographically fair learning: Algorithms and generalization. *arXiv preprint*, 2021. `arXiv:2102.08454`.

9 Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. Convergent algorithms for (relaxed) minimax fairness. *arXiv preprint*, 2020. `arXiv:2011.03108`.

10 Dongliang Xie, Xin Wang, and Linhui Ma. Lexicographical order max-min fair source quota allocation in mobile delay-tolerant networks. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pages 1–6, 2016. `doi:10.1109/IWQoS.2016.7590424`.

11 Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133, 2018.

12 Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, page 325–332, New York, NY, USA, 1996. Association for Computing Machinery. `doi:10.1145/238061.238163`.

13 Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. *arXiv preprint*, 2021. `arXiv:2101.01739`.

14 Ellen L. Hahne. Round-robin scheduling for max-min fairness in data networks. *IEEE Journal on Selected Areas in communications*, 9(7):1024–1039, 1991.

15 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

16 C. Jung, S. Neel, A. Roth, L. Stapleton, and S. Wu. An algorithmic framework for fairness elicitation. *Preprint*, 2020.

17 Christopher Jung, Changhwa Lee, Mallesh M Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. *arXiv preprint*, 2020. `arXiv:2008.08037`.

18 Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. Learning Theory 2003. `doi:10.1016/j.jcss.2004.10.016`.

19 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572. PMLR, 2018.

20 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 100–109, 2019.

21 Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

22 Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020. `arXiv:2006.13114`.

23 Natalie Martinez, Martin Bertran, and Guillermo Sapiro. Minimax Pareto fairness: A multi objective perspective. In *Proceedings of the 37th International Conference on Machine Learning.* Vienna, Austria, PMLR 119, 2020.

24 Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8, 2020.

25 D. Nace and M. Pióro. Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. *IEEE Communications Surveys and Tutorials*, 10, 2008.

26 W. Ogryczak and Warsaw. Lexicographic max-min optimization for efficient and fair bandwidth allocation. *International network optimization conference (INOC)*, January 2007.

27 Wlodzimierz Ogryczak, Hanan Luss, Dritan Nace, and Michał Pióro. Fair Optimization and Networks: Models, Algorithms, and Applications. *Journal of Applied Mathematics*, September 2014. doi:10.1155/2014/340913.

28 B. Radunovic and J. Le Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15(5):1073–1083, 2007. doi:10.1109/TNET.2007.896231.

29 Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair PCA: One extra dimension. In *Advances in Neural Information Processing Systems*, pages 10976–10987, 2018.

30 Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. 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/0e1feae55e360ff05fef58199b3fa521-Paper.pdf.

31 Berk Ustun, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees. In *International Conference on Machine Learning*, pages 6373–6382, 2019.

32 X. Wang, K. Kar, and J. Pang. Lexicographic max-min fair rate allocation in random access wireless networks. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1294–1300, 2006. doi:10.1109/CDC.2006.377233.

33 Congzhou Zhou and N. F. Maxemchuk. Scalable max-min fairness in wireless ad hoc networks. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, and Hua Zhu, editors, *Ad Hoc Networks*, pages 79–93, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

34 Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press, 2003.

A Proofs from Section 5

▶ **Theorem 13** (Lexifairness for Regression). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact, and bounded with diameter D: $\sup_{\theta,\theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is convex and that there exists constants L_M and G such that $L_z(\cdot) \leq L_M$ and $\|\nabla_\theta L_z(\cdot)\|_2 \leq G$, for all data points $z \in \mathcal{Z}$. We have that for any $\ell \leq K$ and any $\alpha \geq 0$, the model $\hat{\theta}_\ell \in \Theta$ output by Algorithm 2 is (ℓ, α)-convex lexicographic fair.*

Proof. We will show that for every round j, the model $\hat{\theta}_j$ computed by our algorithm is (j, α)-convex lexicographic fair, and as a consequence, the very last model $(\hat{\theta}_\ell)$ is (ℓ, α)-convex lexicographic fair. Fix any round $j \leq \ell$. Let $(\theta^t, \eta_j^t, \lambda^t)_{t=1}^T$ be the sequence of plays in the no-regret dynamics of Algorithm 3 in this round. First, note that by the decomposition given in Equation (6), we have

$$\sum_{t=1}^{T} \mathcal{L}_j\left((\theta^t,\eta_j^t),\lambda^t\right) - \min_{\theta\in\Theta,\eta_j\in[0,j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j\left((\theta,\eta_j),\lambda^t\right)$$
$$= \left\{\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta^t,\lambda^t\right) - \min_{\theta\in\Theta}\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta,\lambda^t\right)\right\} + \left\{\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j^t,\lambda^t\right) - \min_{\eta_j\in[0,j\cdot L_M]}\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j,\lambda^t\right)\right\}.$$

In other words, we can decompose the regret of the Learner into two terms: one is the regret of gradient descent plays corresponding to θ, and the other one is the corresponding regret of gradient descent plays for η_j. Note that by Equations (10) and (11) we have the following bounds on the norm of gradients for the Learner. We also use the fact that when the Auditor is best responding, $w_r(\lambda^t)$ can be simplified as in Fact 11.

$$\left\|\nabla_\theta \mathcal{L}_j\left((\theta,\eta_j),\lambda^t\right)\right\|_2 \le \sum_{r=1}^{K}\left|w_r(\lambda^t)\right|\cdot\left\|\nabla_\theta L_r(\theta)\right\|_2 \le jBG$$

$$\left\|\nabla_{\eta_j} \mathcal{L}_j\left((\theta,\eta_j),\lambda^t\right)\right\|_2 = \left|1 - \sum_{\{i_1,\ldots,i_j\}\subseteq[K]} \lambda^t_{\{i_1,i_2,\ldots,i_j\}}\right| \le 1+B$$

Now letting $\eta = \frac{D}{jBG\sqrt{T}}$ and $\eta' = \frac{jL_M}{(1+B)\sqrt{T}}$ in Algorithm 3 and using the regret bound of Online Projected Gradient Desccent (Theorem 18), we have

$$\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta^t,\lambda^t\right) - \min_{\theta\in\Theta}\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta,\lambda^t\right) \le jBGD\sqrt{T}$$

$$\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j^t,\lambda^t\right) - \min_{\eta_j\in[0,j\cdot L_M]}\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j,\lambda^t\right) \le j(B+1)L_M\sqrt{T}$$

and therefore the regret of the Learner can be bounded by

$$\sum_{t=1}^{T} \mathcal{L}_j\left((\theta^t,\eta_j^t),\lambda^t\right) - \min_{\theta\in\Theta,\eta_j\in[0,j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j\left((\theta,\eta_j),\lambda^t\right) \le j(GD+L_M)(B+1)\sqrt{T} := \nu_j T.$$

Let $\nu_j \triangleq j(GD+L_M)(B+1)/\sqrt{T}$. Now using the guarantees of the no-regret dynamics (Theorem 9), the average play of the players $(\hat\theta,\hat\eta_j,\hat\lambda)$ forms a ν_j-approximate equilibrium of the game in the sense that

$$\mathcal{L}_j\left((\hat\theta,\hat\eta_j),\hat\lambda\right) \le \min_{\theta\in\Theta,\eta_j\in[0,j\cdot L_M]} \mathcal{L}_j\left((\theta,\eta_j),\hat\lambda\right) + \nu_j, \quad \mathcal{L}_j\left((\hat\theta,\hat\eta_j),\hat\lambda\right) \ge \max_{\lambda\in\Lambda_j} \mathcal{L}_j\left((\hat\theta,\hat\eta_j),\lambda\right) - \nu_j.$$

Finally, using Theorem 12 we can turn these into the following guarantees. First,

$$\hat\eta_j \le OPT_j\left(\hat\eta_1,\ldots,\hat\eta_{j-1}\right) + 2\nu_j \tag{12}$$

and second, for all $r \le j$,

$$\max_{\{i_1,\ldots,i_r\}\subseteq[K]} \sum_{s=1}^{r} L_{i_r}(\hat\theta_j) \le \hat\eta_r + \frac{jL_M + 2\nu_j}{B}. \tag{13}$$

Define $\epsilon_r \triangleq \hat\eta_r - OPT_r\left(\hat\eta_1,\ldots,\hat\eta_{r-1}\right)$ for all $r \le j$ (ϵ's here are basically *constant* mappings in $\mathbb{R}^{\mathcal{H}}$). We immediately have from Equation (12) that: $\epsilon_r \le 2\nu_r$, for all $r \le j$. Now let $\vec\epsilon = (\epsilon_1,\ldots,\epsilon_j)$, and let $\mathcal{F}^{\vec\epsilon}_{(0)} = \Theta$ be the initial model class. Note that according to Definition 5 and given the defined $\vec\epsilon$, we have for every $r \le j$,

$$\min_{\theta \in \mathcal{F}^{\vec{\epsilon}}_{(r-1)}} \max_{\{i_1,\dots,i_r\}\subseteq[K]} \sum_{s=1}^{r} L_{i_r}(\theta) \equiv OPT_r\left(\hat{\eta}_1,\dots,\hat{\eta}_{r-1}\right).$$

And therefore, by Equation (13), for all $r \leq j$:

$$\begin{aligned}\max_{\{i_1,\dots,i_r\}\subseteq[K]} \sum_{s=1}^{r} L_{i_r}(\hat{\theta}_j) &\leq \hat{\eta}_r + \frac{jL_M + 2\nu_r}{B} \\ &= OPT_r\left(\hat{\eta}_1,\dots,\hat{\eta}_{r-1}\right) + \epsilon_r + \frac{jL_M + 2\nu_r}{B} \\ &= \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(r-1)}} \max_{\{i_1,\dots,i_r\}\subseteq[k]} \sum_{s=1}^{r} L_{i_r}(g) + \epsilon_r + \frac{jL_M + 2\nu_r}{B}\end{aligned}$$

which completes the proof by the choice of $\nu_r = \frac{\alpha}{2}$ for all $r \leq j$ (to guarantee that $\|\vec{\epsilon}\|_\infty \leq \alpha$), and $B = \frac{\alpha + jL_M}{\alpha}$. Note that this setting of parameters, together with $\nu_j = j(GD + L_M)(B+1)/\sqrt{T}$, implies that

$$T = \frac{4j^2(GD + L_M)^2(2\alpha + jL_M)^2}{\alpha^4}.$$ ◀

B Proofs from Section 7

▶ **Theorem 15** (Generalization for Convex Lexifairness). *Fix any distribution $\mathcal{P}$. Suppose for every $\delta > 0$, there exists $\beta(\delta)$ such that the following uniform convergence bound holds.*

$$\Pr_S\left[\max_{h\in\mathcal{H}, k\in[K]} \left|L_k\left(h,S\right) - L_k\left(h,\mathcal{P}\right)\right| > \beta(\delta)\right] < \delta$$

where S is a data set sampled i.i.d. from $\mathcal{P}$. We have that for every data set S sampled i.i.d. from $\mathcal{P}$, if a model h satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1-\delta$ it also satisfies (ℓ, α')-convex lexicographic fairness with respect to $\mathcal{P}$ for $\alpha' = \alpha + 2\ell\beta(\delta)$.

Proof. Fix a distribution $\mathcal{P}$ and a data set S sampled *i.i.d.* from $\mathcal{P}$. Suppose h satisfies (ℓ, α)-convex lexicographic fairness with respect to S. Therefore, according to our convex lexifairness definition, there exists a sequence of mappings $\vec{\epsilon} = (\epsilon_1, \dots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, and a sequence of function classes $\{\mathcal{F}^{\vec{\epsilon}}_{(j)}(S)\}_j$ such that

$$\max_{1\leq j\leq \ell}\left\{\max_{h'\in\mathcal{H}} \epsilon_j(h')\right\} \leq \alpha$$

and that for all $j \leq \ell$:

$$\max_{\{i_1,\dots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h,S) \leq \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\dots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g,S) + \epsilon_j(h) + \alpha \tag{14}$$

where recall that $\mathcal{F}^{\vec{\epsilon}}_{(0)}(S) = \mathcal{H}$ and that for all $j \in [\ell]$,

$$\mathcal{F}^{\vec{\epsilon}}_{(j)}(S) = \left\{h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S) : \max_{\{i_1,\dots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h',S) \leq \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\dots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g,S) + \epsilon_j(h')\right\}.$$

Let us define a mapping $\nu_j^1 : \mathcal{H} \to \mathbb{R}$ such that for every $h' \in \mathcal{H}$,

$$\nu_j^1(h') \triangleq \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) - \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', S)$$

and let

$$\nu_j^2 \triangleq \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, S) - \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P})$$

Now define for every $h' \in \mathcal{H}$, $\tau_j(h') \triangleq \epsilon_j(h') + \nu_j^1(h') + \nu_j^2$ and let $\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P})$ be defined according to our convex lexifairness definition with the sequence of mappings defined by $\vec{\tau} = (\tau_1, \ldots, \tau_\ell)$. In other words, $\mathcal{F}^{\vec{\tau}}_{(0)}(\mathcal{P}) = \mathcal{H}$, and for all $j \in [\ell]$,

$$\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) =$$
$$\left\{ h' \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}) : \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \right\}.$$

▶ **Claim 17.** *For all j, $\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) = \mathcal{F}^{\vec{\epsilon}}_{(j)}(S)$.*

Proof. We use induction on j. For $j = 0$, we have $\mathcal{F}^{\vec{\tau}}_{(0)}(\mathcal{P}) = \mathcal{F}^{\vec{\epsilon}}_{(0)}(S) = \mathcal{H}$. For $j \ge 1$, we have

$$\begin{aligned}
& h' \in \mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) \\
\iff & h' \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}), \\
& \quad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \\
\iff & h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S), \\
& \quad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \\
\iff & h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S), \\
& \quad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', S) \le \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h') \\
\iff & h' \in \mathcal{F}^{\vec{\epsilon}}_{(j)}(S)
\end{aligned}$$

where the second line follows from the induction assumption ($\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}) = \mathcal{F}^{\vec{\vec{\epsilon}}}_{(j-1)}(S)$) and the third line follows from the definition of τ_j. This establishes our claim. ◀

We have that for all $j \le \ell$, the model h satisfies

$$\max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) = \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h, S) + \nu_j^1(h) \le \ldots$$

$$\begin{aligned}
\dots \leq & \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\dots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h) + \alpha + \nu_j^1(h) \\
= & \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\dots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \nu_j^2 + \epsilon_j(h) + \alpha + \nu_j^1(h) \\
= & \min_{g \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\dots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h) + \alpha
\end{aligned}$$

where the first inequality follows from Equation (14). The third line follows from the definition of ν_j^2. The last equality follows from Claim 17 and the fact that $\tau_j(h) = \epsilon_j(h) + \nu_j^1(h) + \nu_j^2$. The proof is complete by the uniform convergence bound provided in the theorem statement. With probability at least $1 - \delta$ over the random draws of the data set S, we have $\max_{h' \in \mathcal{H}} |\nu_j^1(h')| \leq j\beta(\delta)$ and $|\nu_j^2| \leq j\beta(\delta)$, and hence for all $j \leq \ell$,

$$\begin{aligned}
\|\tau\|_\infty = & \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} \tau_j(h') \right\} \\
\leq & \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} \epsilon_j(h') \right\} + \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} |\nu_j^1(h')| + |\nu_j^2| \right\} \\
\leq & \alpha + 2l\beta(\delta).
\end{aligned}$$ ◀

C No-Regret Learning Algorithms

C.1 Online Projected Gradient Descent

Consider an online setting where a learner is playing against an adversary. The learner's action space is some Euclidean subspace $\Theta \subseteq \mathbb{R}^d$ which is equipped with the ℓ_2 norm denoted by $\|\cdot\|_2$. At every round t of the interaction between the learner and the adversary, the learner picks an action $\theta^t \in \Theta$ and the adversary chooses a loss function $\ell^t : \Theta \to \mathbb{R}_{\geq 0}$. The learner then incurs a loss of $\ell^t(\theta^t)$ at that round. Suppose the learner is using some algorithm $\mathcal{A}$ to update its actions from round to round. The goal for the learner is that the regret of $\mathcal{A}$ defined as

$$R_{\mathcal{A}}(T) \triangleq \sum_{t=1}^{T} \ell^t(\theta^t) - \min_{\theta \in \Theta} \sum_{t=1}^{T} \ell^t(\theta)$$

grows sublinearly in T. When Θ and the loss functions played by the adversary are convex, a standard choice of algorithm to use for the learner is *Online Projected Gradient Descent* (Algorithm 4), where in each round, the algorithm updates its action θ^{t+1} for the next round by taking a step in the opposite direction of the gradient of the loss function evaluated at the action of that round: $\nabla \ell^t(\theta^t)$. The updated action is then projected onto the feasible action space Θ: $\text{Proj}_\Theta(\theta) \triangleq \text{argmin}_{\theta' \in \Theta} \|\theta - \theta'\|_2$. Note if the loss functions are not differentiable, we can use subgradients (which are defined given the convexity of the loss functions) instead of gradients and the guarantees will remain.

▶ **Theorem 18** (Regret for Online Projected Gradient Descent [34]). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact and has bounded diameter D: $\sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose for all t, the loss functions ℓ^t are convex and that there exists some G such that $\|\nabla \ell^t(\cdot)\|_2 \leq G$. Let $\mathcal{A}$ be Algorithm 4 run with learning rate $\eta = D/(G\sqrt{T})$. We have that for every sequence of loss functions $(\ell^1, \ell^2, \dots, \ell^T)$ played by the adversary, $R_{\mathcal{A}}(T) \leq GD\sqrt{T}$.*

Algorithm 4 Online Projected Gradient Descent.

```
Input: learning rate $\eta$
Initialize the learner $\theta^1 \in \Theta$;
for $t = 1, 2, \ldots$ do
    Learner plays action $\theta^t$;
    Adversary plays loss function $\ell^t$;
    Learner incurs loss of $\ell^t(\theta^t)$;
    Learner updates its action:
        $\theta^{t+1} = \mathrm{Proj}_\Theta\left(\theta^t - \eta \nabla \ell^t(\theta^t)\right)$
end
```

Algorithm 5 Follow the Perturbed Leader (FTPL).

```
Input: learning rate $\eta$
Initialize the learner $a^1 \in A$;
for $t$ = 1,2, ... do
    Learner plays action $a^t$;
    Adversary plays loss vector $\ell^t$;
    Learner incurs loss of $\langle \ell^t, a^t \rangle$. Learner updates its action:
        $a^{t+1} = \operatorname*{argmin}_{a \in A} \left\{ \left\langle \sum_{s \le t} \ell^s, a \right\rangle + \frac{1}{\eta} \langle \xi^t, a \rangle \right\}$
     where $\xi^t \sim Uniform\left([0,1]^d\right)$, independent of every other randomness.
end
```

C.2 Follow the Perturbed Leader

Here assume the learner's action space is $A \subseteq \{0,1\}^d$. At every round t, the learner chooses an action $a^t \in A$ and then the adversary plays a loss vector $\ell^t \in \mathbb{R}^d$. The learner then incurs a loss of $\langle \ell^t, a^t \rangle$ which is the inner product if a^t and ℓ^t. Suppose the learner is using some algorithm $\mathcal{A}$ to pick its actions in every round. The goal for the learner is to ensure that the regret of $\mathcal{A}$ defined as $R_\mathcal{A}(T) \triangleq \sum_{t=1}^T \langle \ell^t, a^t \rangle - \min_{a \in \mathcal{A}} \sum_{t=1}^T \langle \ell^t, a \rangle$ grows sublinearly in T. *Follow the Perturbed Leader (FTPL)* ([18]), which is described in Algorithm 5, can provide guarantees in this setting.

▶ **Theorem 19** (Regret of FTPL [18]). *Suppose for all t, $\ell^t \in [-M, M]^d$. Let $\mathcal{A}$ be Algorithm 5 run with learning rate $\eta = 1/(M\sqrt{dT})$. We have that for every sequence of loss vectors $(\ell^1, \ell^2, \ldots, \ell^T)$ played by the adversary, $\mathbb{E}\left[R_\mathcal{A}(T)\right] \le 2Md^{3/2}\sqrt{T}$, where expectation is taken with respect to the randomness in $\mathcal{A}$.*

Causal Intersectionality and Fair Ranking

Ke Yang ✉
New York University, NY, USA

Joshua R. Loftus ✉
London School of Economics, UK

Julia Stoyanovich ✉
New York University, NY, USA

Abstract

In this paper we propose a causal modeling approach to intersectional fairness, and a flexible, task-specific method for computing intersectionally fair rankings. Rankings are used in many contexts, ranging from Web search to college admissions, but causal inference for fair rankings has received limited attention. Additionally, the growing literature on causal fairness has directed little attention to intersectionality. By bringing these issues together in a formal causal framework we make the application of intersectionality in algorithmic fairness explicit, connected to important real world effects and domain knowledge, and transparent about technical limitations. We experimentally evaluate our approach on real and synthetic datasets, exploring its behavior under different structural assumptions.

2012 ACM Subject Classification Computing methodologies → Ranking

Keywords and phrases fairness, intersectionality, ranking, causality

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.7

Related Version *Extended Version*: https://arxiv.org/abs/2006.08688

Supplementary Material *Software (Source Code)*: https://github.com/DataResponsibly/CIFRank
archived at swh:1:dir:d16d7e476bfacf0d8c562f2f96b0ead488ad7138

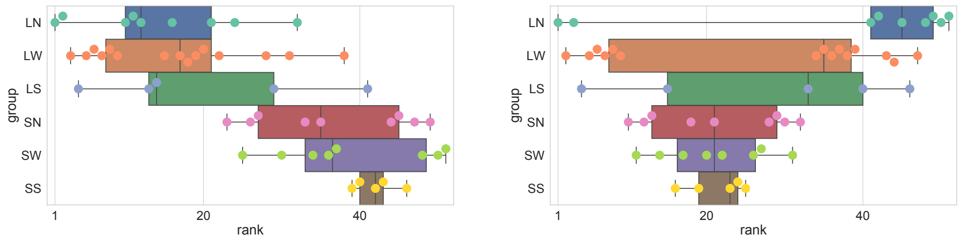
Funding This work was supported in part by National Science Foundation (NSF) Grants No. 1926250, 1934464, and 1916505.

1 Introduction

The machine learning community recognizes several important normative dimensions of information technology including privacy, transparency, and fairness. In this paper we focus on fairness – a broad and inherently interdisciplinary topic of which the social and philosophical foundations are not settled [11]. To connect to these foundations, we take an approach based on *causal modeling*. We assume that a suitable causal generative model is available and specifies relationships between variables including the *sensitive attributes*, which define individual traits or social group memberships relevant for fairness. The model is a statement about how the world works, and we define fairness based on the model itself. In addition to being philosophically well-motivated and explicitly surfacing normative assumptions, the connection to causality gives us access to a growing literature on causal methods in general and causal fairness in particular.

Research on algorithmic fairness has mainly focused on classification and prediction tasks, while we focus on ranking. We consider two types of ranking tasks: score-based and learning to rank (LTR). In score-based ranking, a given set of candidates is sorted on the score attribute (which may itself be computed on the fly) and returned in sorted order. In LTR, supervised learning is used to predict the ranking of unseen items. In both cases, we typically return the highest scoring k items, the top-k. Set selection is a special case of ranking that ignores the relative order among the top-k.

2nd Symposium on Foundations of Responsible Computing (FORC 2021).
Editors: Katrina Ligett and Swati Gupta; Article No. 7; pp. 7:1–7:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) original ranking.

(b) counterfactually fair.

Figure 1 CSRanking by weighted publication count, showing positions of intersectional groups by department size, large (L) and small (S), and location, North East (N), West (W), South East (S). Observe that the top-20 in Figure 1a is dominated by large departments, particularly those from the West and from the North East. Treating small departments from the South East as the disadvantaged intersectional group, and applying the techniques described in Section 2 of the paper, we derive the ranking in Figure 1b that has more small department at the top-20 and is more geographically balanced.

Further, previous research mostly considered a single sensitive attribute, while we use multiple sensitive attributes for the fairness component. As noted by Crenshaw [14], it is possible to give the appearance of being fair with respect to each sensitive attribute such as race and gender separately, while being unfair with respect to *intersectional* subgroups. For example, if fairness is taken to mean proportional representation among the top-k, it is possible to achieve proportionality for each gender subgroup (e.g., men and women) and for each racial subgroup (e.g., Black and White), while still having inadequate representation for a subgroup defined by the intersection of both attributes (e.g., Black women). The literature on intersectionality includes theoretical and empirical work showing that people adversely impacted by more than one form of structural oppression face additional challenges in ways that are more than additive [12, 16, 37, 43].

1.1 Contribution

We define intersectional fairness for ranking in a similar manner to previous causal definitions of fairness for classification or prediction tasks [10, 26, 30, 36, 55]. The idea is to model the causal effects between sensitive attributes and other variables, and then make algorithms fairer by removing these effects. With a given ranking task, set of sensitive attributes, and causal model, we propose ranking on counterfactual scores as a method to achieve intersectional fairness. From the causal model we compute model-based counterfactuals to answer a motivating question like "What would this person's data look like if they had (or had not) been a Black woman (for example)?" We compute counterfactual scores *treating every individual in the sample as though they had belonged to one specific, baseline intersectional subgroup*. For score-based ranking we then rank these counterfactual scores, but the same approach to causal intersectional fairness can be combined with other machine learning tasks, including prediction (not necessarily specific to ranking).

The choice of a baseline counterfactual subgroup is essentially arbitrary, and there are other possibilities like randomizing or averaging over all subgroups. We focus on using one subgroup now for simplicity, but in principle this choice can depend on problem specifics and future work can investigate dependence on this choice. In fact, our framework allows for numeric sensitive attributes, like age for example, where treating everyone according to one baseline counterfactual is possible even though subgroup terminology breaks down. In this case we can still try to rank every individual based on an answer to a motivating question like "What would this person's data look like if they were a 45-year old Black woman?"

While intersectional concerns are usually raised when data is about people, they also apply for other types of entities. Figure 1 gives a preview of our method on the CSRankings dataset [5] that ranks 51 computer science departments in the US by a weighted publication count score (lower ranks are better). Departments are of two sizes, large (L, with more than 30 faculty members) and small (S), and are located in three geographic areas, North East (N), West (W), and South East (S). The original ranking in Figure 1a prioritizes large departments, particularly those in the North East and in the West. The ranking in Figure 1b was derived using our method, treating small departments from the South East as the disadvantaged intersectional group; it includes small departments at the top-20 and is more geographically balanced.

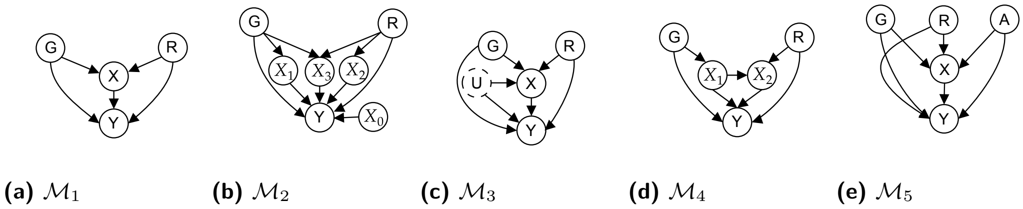
We begin with relatively simple examples to motivate our ideas before considering more complex ones. The framework we propose can, under the right conditions, disentangle multiple interlocked "bundles of sticks," to use the metaphor in Sen and Wasow [42] for causally interpreting sensitive attributes that may be considered immutable. We see this as an important step towards a more nuanced application of causal modeling to fairness.

1.2 Motivating example: Hiring by a moving company

Consider an idealized hiring process of a moving company, inspired by Datta et al. [15], in which a dataset of applicants includes their gender G, race R, weight-lifting ability score X, and overall qualification score Y. A ranking of applicants $\boldsymbol{\tau}$ sorts them in descending order of Y. We assume that the structural causal model shown in Figure 2a describes the data generation process, and our goal is to use this model to produce a ranking that is fair with respect to race, gender, and the intersectional subgroups of these categories. The arrows in the graph pointing from G and R directly to Y represent the effect of "direct" discrimination. Under US labor law, the moving company may be able to make a "business necessity" argument [17] that they are not responsible for any "indirect" discrimination on the basis of the mediating variable X. If discrimination on the basis of X is considered unenforceable, we refer to X as a *resolving mediator*, and denote this case as the resolving case, following the terminology of Kilbertus et al. [26].

A mediator X may be considered resolving or not; this decision can be made separately for different sensitive attributes, and the relative strengths of causal influences of sensitive attributes on both X and Y can vary, creating potential for explanatory nuance even in this simple example. Suppose that X is causally influenced by G but not by R, or that the relative strength of the effect of G on X is larger than that of R. Then, if X is considered resolving, the goal is to remove direct discrimination on the basis of both R and G, but hiring rates might still differ between gender groups if that difference is explained by each individual's value of X. On the other hand, if X is not considered resolving, then the goal also includes removing indirect discrimination through X, which, in addition to removing direct discrimination, might accomplish positive discrimination, in the style of affirmative, action based on the effect of G on X.

Once the goal has been decided, we use the causal model to compute counterfactual scores Y – the scores that would have been assigned to the individuals if they belonged to one particular subgroup defined by fixed values of R and G, while holding the weight-lifting score X fixed in the resolving case – and then rank the candidates based on these scores. The moving company can then interview or hire the highly ranked candidates, and this process would satisfy a causal and intersectional definition of fairness. We analyze a synthetic dataset based on this example in Section 3 with results shown in Figure 3a.



(a) $\mathcal{M}_1$ **(b)** $\mathcal{M}_2$ **(c)** $\mathcal{M}_3$ **(d)** $\mathcal{M}_4$ **(e)** $\mathcal{M}_5$

Figure 2 Causal models that include sensitive attributes G (gender), R (race), and A (age), utility score Y, other covariates $\mathbf{X}$, and a latent (unobserved) variable U.

1.3 Organization of the paper

In Section 2 we introduce notation and describe the particular causal modeling approach we take, using directed acyclic graphs and structural equations, but we also note that our higher level ideas can be applied with other approaches to causal modeling. We present the necessary modeling complexity required for interaction effects in the causal model, the process of computing counterfactuals for both the resolving and non-resolving cases, and the formal fairness definition that our process aims to satisfy. In Section 3 we demonstrate the effectiveness of our method on real and synthetic dataset. We present a non-technical interpretation of our method, and discuss its limitations, in Section 4. We summarize related work in Section 5 and conclude in Section 6. Our code is publicly available at `https://github.com/DataResponsibly/CIFRank`.

2 Causal intersectionality

In this section we describe the problem setting, and present our proposed definition of intersectional fairness within causal models and an approach to computing rankings satisfying the fairness criterion.

2.1 Model and problem setting

2.1.1 Causal model

As an input, our method requires a structural causal model (SCM), which we define briefly here and refer to [23, 33, 39, 44] for more detail. An SCM consists of a directed acyclic graph (DAG) $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where the vertex set $\mathbf{V}$ represents variables, which may be observed or latent, and the edge set $\mathbf{E}$ indicates causal relationships from source vertices to target vertices. Several example DAGs are shown in Figure 2, where vertices with dashed circles indicate latent variables.

For $V_j \in \mathbf{V}$ let $\text{pa}_j = \text{pa}(V_j) \subseteq \mathbf{V}$ be the "parent" set of all vertices with a directed edge into V_j. If pa_j is empty, we say that V_j is exogenous, and otherwise we assume that there is a function $f_j(\text{pa}_j)$ that approximates the expectation or some other link function, such as the logit, of V_j. Depending on background knowledge or the level of assumptions we are willing to hazard, we assume that functions f_j are either known or can be estimated from the data. We also assume a set of sensitive attributes $\mathbf{A} \subseteq \mathbf{V}$, chosen *a priori*, for which existing legal, ethical, or social norms suggest that the ranking algorithm should be fair.

2.1.2 Problem setting

In most of our examples we consider two sensitive attributes, which we denote G and R, motivated by the example of Crenshaw [14] of gender and race. We let Y denote an outcome variable that is used as a utility score in our ranking task, and $\mathbf{X}$ be *a priori* non-sensitive predictor variables. In examples with pathways from sensitive attributes to Y passing through $\mathbf{X}$ we call the affected variables in $\mathbf{X}$ mediators. Finally, U may denote an unobserved confounder. In some settings a mediator may be considered *a priori* to be a legitimate basis for decisions even if it results in disparities. This is what Foulds et al. [18] call the infra-marginality principle, others [10, 30, 36] refer to as path-specific effects, and Zhang and Bareinboim [55] refer to as indirect effects; Kilbertus et al. [26] call such mediators *resolving variables*. We adopt the latter terminology and will show examples of different cases later. In fact, our method allows mediators to be resolving for one sensitive attribute and not for the other, reflecting nuances that may be necessary in intersectional problems.

For simplicity of presentation, we treat some sensitive attributes as binary indicators of a particular privileged status, rather than using a more fine grained coding of identity, but note that this is not a necessary limitation of the method. Our experiments in Section 3 use models $\mathcal{M}_1$ in Figure 2a and $\mathcal{M}_5$ in Figure 2e, but richer datasets and other complex scenarios such as $\mathcal{M}_2$ also fit into our framework. *Sequential ignorability* [21, 38, 40, 47] is a standard assumption for model identifiability that can be violated by unobserved confounding between a mediator and an outcome, as in $\mathcal{M}_3$ in Figure 2c, or by observed confounding where one mediator is a cause of another, as in $\mathcal{M}_4$ in Figure 2d. We include these as indications of qualitative limitations of this framework.

2.2 Counterfactual intersectional fairness

2.2.1 Intersectionality

It is common in predictive modeling to assume a function class that is linear or additive in the inputs, that is, for a given non-sensitive variable V_j:

$$f_j(\mathrm{pa}_j) = \sum_{V_l \in \mathrm{pa}_j} f_{j,l}(V_l).$$

Such simple models may be less likely to overfit and are more interpretable. However, to model the intersectional effect of multiple sensitive attributes *we must avoid this assumption.* Instead, we generally assume that f_j contains non-additive interactions between sensitive attributes. With rich enough data, such non-linear f_j can be modeled flexibly, but to keep some simplicity in our examples we will consider functions with linear main effects and second order interactions. That is, if the set pa_j of parents of V_j includes q sensitive attributes $A_{j_1}, A_{j_2}, \ldots, A_{j_q}$ and p non-sensitive attributes $X_{j_{q+1}}, X_{j_{q+2}}, \ldots X_{j_{q+p}}$, we assume

$$f_j(\mathrm{pa}_j) = \beta_0^{(j)} + \sum_{l=1}^{p} \beta_l^{(j)} X_{j_{q+l}} + \sum_{l=1}^{q} \eta_l^{(j)} A_{j_l} + \sum_{l=1}^{q-1} \sum_{r=l+1}^{q} \eta_{r,l}^{(j)} A_{j_l} A_{j_r}. \tag{1}$$

The coefficients (or weights) $\eta_l^{(j)}$ model the main causal effect on V_j of disadvantage on the basis of sensitive attribute A_{j_l}, while $\eta_{r,l}^{(j)}$ model the non-additive combination of adversity related to the interactions of A_{j_r} and A_{j_l}. For the example the model $\mathcal{M}_1$ in Figure 2a with sensitive attributes G and R, mediator X, and outcome Y, we can write (1) for Y as

$$f_Y(X, G, R) = \beta_0^{(Y)} + \beta_1^{(Y)} X + \eta_G^{(Y)} G + \eta_R^{(Y)} R + \eta_{R,G}^{(Y)} RG \tag{2}$$

For ease of exposition we mostly focus on categorical sensitive attributes, and in that case (1) can be reparameterized with a single sensitive attribute with categories for each intersectional subgroup. In the simplest cases then it may appear this mathematical approach to intersectional fairness reduces to previously considered fairness problems. However, our framework is not limited to the simplest cases. And even with two binary sensitive attributes it may be necessary to model the separate causal relationships between each of these and one or more mediators, which may also be considered resolving or non-resolving separately with respect to each sensitive attribute. With numeric attributes our framework can include non-linear main effects and higher order interactions, and in Appendix A.2 we present results for an experiment with a numeric sensitive attribute.

Our experiments use simpler examples with one mediator so the results are easier to interpret and compare to non-causal notions of fairness in ranking. Sophisticated models like Figure 2b, with combinations of resolving and non-resolving mediators, would be more difficult to compare to other approaches, but we believe this reflects that real-world intersectionality can pose hard problems that our framework is capable of analyzing. And while identifiability and estimation are simplified in binary examples, the growing literature on causal mediation discussed in Section 5 can be used on harder problems.

2.2.2 Counterfactuals

Letting $\mathbf{A}$ denote the vector of sensitive attributes and $\mathbf{a}'$ any possible value for these, we compute the counterfactual $Y_{\mathbf{A}\leftarrow\mathbf{a}'}$ by replacing the observed value of $\mathbf{A}$ with $\mathbf{a}'$ and then propagating this change through the DAG: any directed descendant V_j of $\mathbf{A}$ has its value changed by computing $f_j(\mathrm{pa}_j)$ with the new value of $\mathbf{a}'$, and this operation is iterated until it reaches all the terminal nodes that are descendants of any of the sensitive attributes $\mathbf{A}$. We interpret these model-based counterfactuals informally as "the value Y would have taken if $\mathbf{A}$ had been equal to $\mathbf{a}'$."

For graphs with resolving mediators we may keep the mediator fixed while computing counterfactuals. We describe this process in detail for model $\mathcal{M}_1$ in Figure 2a, with both the resolving and the non-resolving cases. We focus on this model for clarity, but all that we say in the rest of this section requires only minor changes to hold for other models such as $\mathcal{M}_2$ without loss of generality, provided they satisfy sequential ignorability [21, 38, 40, 47]. Our implementation is similar to what Kusner et al. [30] refer to as "Level 3" assumptions, but we denote exogenous error terms as ϵ instead of U.

We consider the case where Y is numeric and errors are additive

$$X = f_X(G, R) + \epsilon^X, \quad Y = f_Y(X, G, R) + \epsilon^Y.$$

with f_Y given in (2) and f_X defined similarly. The case where Y is not continuous fits in the present framework with minor modifications, where we have instead a probability model with corresponding link function g so that

$$\mathbb{E}[Y|X, G, R] = g^{-1}(f_Y(X, G, R)).$$

Suppose that the observed values for observation i are (y_i, x_i, g_i, r_i), with exogenous errors $\epsilon_i^X, \epsilon_i^Y$. Since we do not model any unobserved confounders in model $\mathcal{M}_1$, we suppress the notation for U and denote counterfactual scores, for some $(g', r') \neq (g, r)$, as:

$$Y_i' := (Y_i)_{\mathbf{A}\leftarrow\mathbf{a}'} = (Y_i)_{(G,R)\leftarrow(g',r')}.$$

If X is **non-resolving**, then we first compute counterfactual X as $x_i' := f_X(g', r') + \epsilon_i^X$, substituting (g', r') in place of the observed (g_i, r_i). Then we do the same substitution while computing:

$$Y_i' = f_Y(x_i', g', r') + \epsilon_i^Y = f_Y(f_X(g', r') + \epsilon_i^X, g', r') + \epsilon_i^Y .$$

If X is **resolving**, then we keep the observed X and compute:

$$Y_i' = f_Y(x_i, g', r') + \epsilon_i^Y .$$

If X is **semi-resolving**, for example resolving for R but not for G, in which case we compute counterfactual X as $x_i' := f_X(g', r_i) + \epsilon_i^X$ and then

$$Y_i' = f_Y(f_X(g', r_i) + \epsilon_i^X, g', r') + \epsilon_i^Y .$$

If the functions f_X, f_Y have been estimated from the data, then we have observed residuals r_i^X, r_i^Y instead of model errors in the above. Finally, in cases where we model unobserved confounders U we may also attempt to integrate over the estimated distribution of U as described in [30].

2.3 Counterfactually fair ranking

2.3.1 Ranking task

We use an outcome or utility score Y to rank a dataset $\mathbf{D}$, assumed to be generated by a model $\mathcal{M}$ from among the example SCMs in Figure 2. If the data contains a mediating predictor variable X, then the task also requires specification of the resolving status of X. Letting $n = |\mathbf{D}|$, a ranking is a permutation $\boldsymbol{\tau} = \hat{\boldsymbol{\tau}}(\mathbf{D})$ of the n individuals or items, usually satisfying:

$$Y_{\boldsymbol{\tau}(1)} \geq Y_{\boldsymbol{\tau}(2)} \geq \cdots \geq Y_{\boldsymbol{\tau}(n)}. \tag{3}$$

To satisfy other objectives, like fairness, we generally output a ranking $\hat{\boldsymbol{\tau}}$ that is not simply sorting on the observed values of Y. Specifically, we aim to compute counterfactually fair rankings.

▶ **Definition 1** (Counterfactually fair ranking). *A ranking $\hat{\boldsymbol{\tau}}$ is counterfactually fair if, for all possible x and pairs of vectors of actual and counterfactual sensitive attributes $a \neq a'$, respectively, we have:*

$$\begin{aligned} &\mathbb{P}(\hat{\boldsymbol{\tau}}(Y_{\mathbf{A}\leftarrow\mathbf{a}}(U)) = k \mid \mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a}) \\ = &\mathbb{P}(\hat{\boldsymbol{\tau}}(Y_{\mathbf{A}\leftarrow\mathbf{a}'}(U)) = k \mid \mathbf{X} = \mathbf{x}, \mathbf{A} = \mathbf{a}) \end{aligned} \tag{4}$$

for any rank k, and with suitably randomized tie-breaking. If any mediators are considered resolving then the counterfactual $Y_{\mathbf{A}\leftarrow\mathbf{a}'}(U)$ in this definition is computed accordingly, holding such mediators fixed.

This definition is one natural adaptation of causal definitions in the recent literature on fairness in classification and prediction tasks [10, 26, 30, 36, 55] to the ranking setting. To satisfy Equation 4, we rank using counterfactuals that treat all individuals or items in the dataset according to one fixed baseline value $\mathbf{a}'$.

There are other possible definitions relaxing (4), for example using expected rank or enforcing equality for some but not all values of k. We leave the problems of deriving algorithms satisfying these and comparing performance to future work.

2.3.2 Implementation

We use the following procedure to compute counterfactually fair rankings, keeping our focus on model $\mathcal{M}_1$ in Figure 2a for clarity and readability.

1. For a (training) dataset $\mathbf{D}$, we estimate the parameters of the assumed causal model $\mathcal{M}$. A variety of frequentist or Bayesian approaches for estimation can be used. Our experiments use the R package `mediation` [46] on model $\mathcal{M}_1$ in Figure 2a.
2. From the estimated causal model we compute counterfactual records on the (training) data, transforming each observation to one reference subgroup $\mathbf{A} \leftarrow \mathbf{a}'$, we set $\mathbf{a}'$ to be the disadvantaged intersectional group. This yields counterfactual training data $\mathbf{D}_{\mathbf{A}\leftarrow\mathbf{a}'}$.
3. For score-based ranking, we sort $Y_{\mathbf{A}\leftarrow\mathbf{a}'}$ in descending order to produce the counterfactually fair ranking $\hat{\tau}(Y_{\mathbf{A}\leftarrow\mathbf{a}'})$. For learning to rank (LTR), we apply a learning algorithm on $\mathbf{D}_{\mathbf{A}\leftarrow\mathbf{a}'}$ and consider two options, depending on whether the problem structure allows the use of the causal model at test time: if it does, then we in-process the test data from the learned causal model before ranking counterfactual test scores, and if it does not, then we rank the unmodified test data. We refer to the first case as cf-LTR and emphasize that in the second case *counterfactually fairness may not hold, or hold only approximately, on test data.*

Proposition 2 below says that this implementation, under common causal modeling assumptions, satisfies our fair ranking criteria. The proof is in Appendix A.1.

▶ **Proposition 2** (Implementing counterfactually fair ranking). *If the assumed causal model $\mathcal{M}$ is identifiable and correctly specified, implementations described above produce counterfactually fair rankings in the score-based ranking and cf-LTR tasks.*

3 Experimental Evaluation

In this section we investigate the behavior of our framework under different structural assumptions of the underlying causal model on real and synthetic datasets. We quantify performance with respect to several fairness and utility measures, for both score-based rankers and for learning to rank.

3.1 Datasets and evaluation measures

Datasets

We present experimental results on the real dataset COMPAS [1] and on a synthetic benchmark that simulates hiring by a moving company, inspired by Datta et al. [15]. We also present results on another synthetic benchmark that is a variant of the moving company dataset, but with an additional numerical sensitive attribute, in Appendix A.2.

COMPAS contains arrest records with sensitive attributes gender and race. We use a subset of *COMPAS* that includes Black and White individuals of either gender with at least 1 prior arrest. The resulting dataset has 4,162 records with about 25% White males, 59% Black males, 6% White females, and 10% Black females. We fit the causal model $\mathcal{M}_1$ in Figure 2a with gender G, race R, number of prior arrests X, and COMPAS decile score Y, with larger Y predicting higher likelihood of recidivism. In our use of this dataset, we will rank defendants on Y from lower to higher, prioritizing them for release or for access to supportive services as part of a comprehensive reform of the criminal justice system.

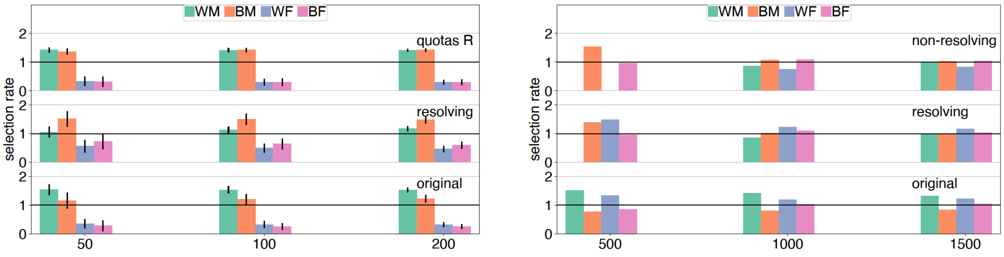
Moving company is a synthetic dataset drawn from the causal model $\mathcal{M}_1$ in Figure 2a, with edge weights: $w(G \rightarrow X) = 1$, $w(R \rightarrow X) = 0$, $w(G \rightarrow Y) = 0.12$, $w(R \rightarrow Y) = 0.08$, and $w(X \rightarrow Y) = 0.8$. This dataset is used in the scenario we discussed in our motivating example in Section 1.2: Job applicants are hired by the moving company based on their qualification score Y, computed from weight-lifting ability score X, and affected by gender G and race R, either directly or through X. Specifically, weight-lifting ability X is lower for female applicants than for male applicants; qualification score Y is lower for female applicants and for Blacks. Thus, the intersectional group Black females faces greater discrimination than either the Black or the female group. In our experiments in this section, we assume that women and Blacks each constitute a 37% minority of the applicants, and that gender and race are assigned independently. As a result, there are about 40% White males, 14% Black females, and 23% of both Black males and White females in the input with $2,000$ records.

Fairness measures

We investigate whether the counterfactual ranking derived using our method is fair with respect to intersectional groups of interest, under the given structural assumptions of the underlying causal model. We consider two types of fairness measures: those that compare ranked outcomes across groups, and those that compare ranked outcomes within a group. To quantify fairness across groups, we use two common measures of fairness in classification that also have a natural interpretation for rankings: *demographic parity (DP) at top-k* and *equal opportunity (EO) at top-k*, for varying values of k. To quantify fairness within a group, we use a rank-aware measure called *in-group fairness ratio (IGF-Ratio)*, proposed by Yang et al. [49] to surface intersectional fairness concerns in ranking. We report our IGF-Ratio results in Appendix A.3, and refer the reader to an extended version of this paper [50] for experiments with other rank-aware fairness measures.

Demographic parity (DP) is achieved if the proportion of the individuals belonging to a particular group corresponds to their proportion in the input. We will represent DP by showing selection rates for each intersectional group at the top-k, with a value of 1 for all groups corresponding to perfect DP.

Equal opportunity (EO) in binary classification is achieved when the likelihood of receiving the positive prediction for items whose true label is positive does not depend on the values of their sensitive attributes [19]. To measure EO for LTR, we will take the set of items placed at the top-k in the ground-truth ranking to correspond to the positive class *for that value of k*. We will then present sensitivity (true positives / true positives + false negatives) per intersectional group at the top-k. If sensitivity is equal for all groups, then the method achieves EO.



(a) *moving company.*

(b) *COMPAS.*

Figure 3 Demographic parity on the *moving company* and *COMPAS* datasets. X-axis shows the top-k values of the rankings and Y-axis shows the selection rate while each span of Y-axis represents different rankings and each color represents an intersectional group. The assumed causal model for both *moving company* and *COMPAS* is $\mathcal{M}_1$ in Figure 2a.

Utility measures

When the distribution of scores Y differs across groups, then we may need to sacrifice score-utility to produce a fair ranking. We evaluate the score-utility of the counterfactual rankings using two measures, *Y-utility loss at top-k*, applicable for both score-based ranking and LTR, and *average precision (AP)*, applicable only for LTR. Both compare a "ground truth" ranking $\boldsymbol{\tau}$ induced by the order of the observed scores Y to a proposed fair ranking $\boldsymbol{\sigma}$ (we use $\boldsymbol{\sigma}$ rather than $\hat{\boldsymbol{\tau}}$ here to make notation more readable).

We define Y-utility loss at top-k as $L_k(\boldsymbol{\sigma}) = 1 - \sum_{i=1}^{k} Y_{\boldsymbol{\sigma}(i)} / \sum_{i=1}^{k} Y_{\boldsymbol{\tau}(i)}$. $Y_{\boldsymbol{\sigma}(i)}$ is the observed score of the item that appears at position i in $\boldsymbol{\sigma}$, while $Y_{\boldsymbol{\tau}(i)}$ is the observed score of the item at position i in the original ranking $\boldsymbol{\tau}$. L_k ranges between 0 (best) and 1 (worst).

Average precision (AP) quantifies, in a rank-compounded manner, how many of the items that should be returned among the top-k are indeed returned. Recall that $\boldsymbol{\tau}_{1\ldots k}$ denotes the *set* of the top-k items in a ranking $\boldsymbol{\tau}$. We define precision at top-k as $P_k = |\boldsymbol{\tau}_{1\ldots k} \cap \boldsymbol{\sigma}_{1\ldots k}|/k$, where $\boldsymbol{\tau}$ is the "ground truth" ranking and $\boldsymbol{\sigma}$ is the predicted ranking. Then, $AP_k(\boldsymbol{\sigma}) = \sum_{i=1}^{k} P_i \times \mathbb{1}[\boldsymbol{\sigma}(i) \in \boldsymbol{\tau}_{1\ldots k}]/k$, where $\mathbb{1}$ is an indicator function that returns 1 if the condition is met and 0 otherwise. AP_k ranges between 0 (worst) and 1 (best).

3.2 Score-based ranking

In the first set of experiments, we focus on score-based rankers, and quantify performance of our method in terms of demographic parity (Figure 3 and 5) and score-based utility, on *moving company* (over 100 executions) and *COMPAS*.

Synthetic datasets

Recall that, in the *moving company* example, the goal is to compute a ranking of the applicants on their qualification score Y that is free of racial discrimination, while allowing for a difference in weight-lifting ability X between gender groups, thus treating X as a resolving variable. Figure 3a compares DP of three rankings for the moving company example: *original*, *resolving*, and *quotas on R*, described below.

Recall that perfect DP is achieved when selection rate equals to 1 for all groups. We observe that the *original* ranking, the bottom set of bars in Figure 3a, under-represents women (WF and BF) compared to their proportion in the input, and that White men (WM) enjoy a higher selection rate than do Black men (BM). Specifically, there are between 62-64% White men (40% in the input), 27-28% Black men (23% in the input), 6% White women (23% in the input), and 3-9% Black women (14% in the input) for $k = 50, 100, 200$.

In comparison, in the counterfactually fair ranking in which X is treated as *resolving*, shown as the middle set of bars in Figure 3a, selection rates are higher for the Blacks of both genders than for the Whites. For example, selection rate for White men is just over 1, while for Black men it's 1.5. Selection rates also differ by gender, because weight-lifting ability X is a mediator, and it encodes gender differences.

Finally, the ranking *quotas R*, the top set of bars in Figure 3a, shows demographic party for racial groups when the ranking is computed using representation constraints (quotas) on race R. This ranking is computed by independently sorting Black and White applicants on Y and selecting the top individuals from each list in proportion to that group's representation in the input. Opting for quotas on race rather than on gender, or on a combination of gender and race, is reasonable here, and it implicitly encodes a normative judgement that is explicit in our causal model $\mathcal{M}_1$ in Figure 2a – that race should not impact the outcome, while gender may.

Appendix A.2 describes another synethetic dataset, *moving company+age*, with three sensitive attributes: categorical gender G and race R, and numerical age A, with records drawn from the causal model $\mathcal{M}_5$ in Figure 2e. Our results on this dataset further showcase the flexibility of our framework.

Real datasets

We now present results of an evaluation of our method on a real dataset, *COMPAS*. Figure 3b shows demographic parity (DP) of three different rankings: *original*, *resolving*, and *non-resolving*, discussed below. Recall that in our use of *COMPAS* defendants are ranked on their decile score Y from lower to higher, prioritizing them for release or for access to supportive services. Our goal is to produce a ranking that is free of racial and gender discrimination. There is some debate about whether the number of prior arrests, X, should be treated as a resolving variable. By treating X as non-resolving, we are stating that the number of prior arrests is itself subject to racial discrimination.

We observe that, in the *original* ranking, shown as the bottom set of bars in Figure 3b, Whites of both genders are selected at much higher rates than Blacks. Gender has different effect by race: men are selected at higher rates for Whites, and at lower rates for Blacks. There are 33-38% White men (25% in the input), 46-49% Black men (59% in the input), 7-8% White women (6% in the input), and 8-10% Black women (10% in the input), for $k = 500, 1000, 1500$.

Comparing the original ranking to the counterfactually fair ranking that treats the number of prior arrests X as a *resolving* mediator, shown as the middle set of bars in Figure 3b, we observe an increase in selection rates for Black males and Black females, and a significant reduction in selection rates for White males. Further, comparing with the counterfactually fair ranking that treats X as *non-resolving*, the top set of bars in Figure 3b, we observe that only Black individuals are represented at the top-500, and that selection rates for all intersectional groups for larger values of k are close to 1, achieving demographic parity.

We also computed utility loss at top-k, based on the original Y scores (see Section 3.1 for details). For *moving company*, we found that counterfactually fair ranking *resolving* suffers at most 1% loss across the values of k, slightly higher than the loss of the *quotas* R ranking, which is close to 0. For *COMPAS*, we found that overall utility loss is low in most cases, ranging between 3% and 8% in the fair ranking *resolving*, and between 3% and 10% in the fair ranking *non-resolving*. The slightly higher loss for the latter case is expected, because we are allowing the model to correct for historical discrimination in the data more strongly in this case, thus departing from the original ranking further.

3.3 Learning to rank

We now investigate the usefulness of our method for supervised learning of counterfactually fair ranking models. We use ListNet, a popular Learning to Rank algorithm, as implemented by Ranklib[1]. ListNet is a listwise method – it takes ranked lists as input and generates predictions in the form of ranked lists. We choose ListNet because of its popularity and effectiveness (see additional information about ListNet and other predictive ranking models in [32] and [34], respectively).

We conduct experiments in two regimes that differ in whether to apply our method as a preprocessing fairness intervention on the test set (see Implementation in Section 2). In both regimes, we make the training datasets counterfactually fair. Specifically, we first fit a causal

[1] https://sourceforge.net/p/lemur/wiki/RankLib/

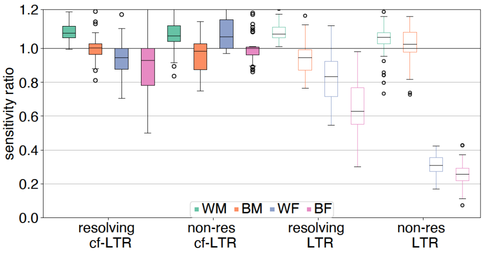


Figure 4 Equal opportunity on *moving company* with $k = 200$. X-axis shows the treatments: training & test on fair rankings with X as resolving (resolving cf-LTR) and non-resolving (non-resolving cf-LTR); training on fair rankings & test on unmodified rankings with X as resolving (resolving LTR) and non-resolving (non-resolving LTR). Y-axis shows the ratio of sensitivity between each counterfactually fair treatment and the original ranking. Intersectional groups are denoted by different colors. Solid boxes correspond to cf-LTR variants. All results are over 50 training/test pairs.

model $\mathcal{M}$ on the training data, then update the training data to include counterfactually fair values of the score Y and of any non-resolving mediators X, and finally train the ranking model $\mathcal{R}$ (e.g., ListNet) on the fair training data. We now have two options: (1) to run $\mathcal{R}$ on the *unmodified (biased) test data*, called *LTR* in our experiments, or; (2) to *preprocess test data* using $\mathcal{M}$, updating test with counterfactually fair values for the score Y and for any non-resolving mediators X, before passing it on to $\mathcal{R}$, called *cf-LTR*.

Note that the cf-LTR setting shows the effectiveness of our method for the disadvantaged intersectional groups, in that the performance of the model is compareble across groups, while LTR setting shows the performance of a ranking model on biased test data. Similar to score-based ranking, we also consider two structural assumptions of the underlying causal model: resolving and non-resolving for each setting above.

We quantify performance of our method in terms of equal opportunity (EO) and average precision (AP) (see Section 3.1), on *moving company* over 50 training/test pairs. Figure 4 shows performance of the ranking model (e.g., ListNet) in terms of equal opportunity on *moving company*, comparing four settings produced from above options: resolving cf-LTR, non-resolving cf-LTR, resolving LTR, and non-resolving LTR. Recall that a method achieves equal opportunity (EO) if sensitivity is equal across groups. Note that sensitivity is affected by groups' representation in the data, meaning that higher sensitivity for a group might be due to its limited representation in the top-k rankings (lower positives) rather than the better treatment in the model (higher true positives). Thus, to reduce the effect of imbalanced representation across groups, we present *sensitivity ratio*: the ratio of the sensitivity at each setting above (with the fairness treatment on training, or on both training and test data) to the sensitivity of the original ranking model (without any fairness intervention) in Figure 4.

Note that the original ranking model achieves high sensitivity for all intersectional groups (0.9, 0.9, 0.95, and 1 for White men, Black men, White women, and Black women, respectively) and so can be seen as achieving EO within gender groups, because their representation at the top-k is similar. As shown in Figure 4, performance of the fair ranking models (e.g., the cf-LTR variants in the left two columns for resolving and non-resolving X respectively), in which both the training and the test data are counterfactually fair, is comparable to the original ranking model in terms of sensitivity, with the medians of all boxes close to the sensitivity ratio of 1.

The resolving variants (e.g., resolving cf-LTR and LTR columns in Figure 4) show lower sensitivity for women, likely because women are selected at lower rates since X is treated as resolving for gender). The LTR variants (e.g., resolving and non-res LTR columns in Figure 4) show lower sensitivity for women because the test dataset is unmodified in this set of experiments. Finally, when the fairness intervention is applied on both training and test datasets (e.g., resolving and non-res cf-LTR columns in Figure 4), it leads to better sensitivity for women.

We also quantified utility as average precision (AP) in evaluating supervised learning of counterfactually fair ranking models. For *moving company*, AP is 77% for the original ranking model when unmodified ranking are used for training and test. For counterfactually fair training data with non-resolving X (weight-lifting), AP on unmodified test (non-res LTR) is 27% but it increases to 91% when test data is preprocessed (non-res cf-LTR). For counterfactually fair training data with resolving X, AP is 68% for unmodified test (resolving LTR) and 83% when test is preprocessed (resolving cf-LTR).

4 Discussion

This work aims to mitigate the negative impacts of ranking systems on people due to attributes that are out their control. In this section we anticipate and discuss concerns that may arise in the application of our method.

There are objections to modeling sensitive attributes as causes rather than considering them to be immutable, defining traits. Some of these objections and responses to them are discussed in [33]. In the present work we proceed with an understanding that the model is a simplified and reductive approximation, and support for deploying an algorithm and claiming it is fair should require an inclusive vetting process where formal models such as these are *tools for inclusively achieving consensus* and not for rubber stamping or obfuscation.

There are many issues outside the scope of the present work but which are important in any real application. Choices of which attributes are sensitive, which mediators are resolving (and for which sensitive attributes), the social construction and definitions of sensitive attributes, choices of outcome/utility or proxies thereof, technical limitations in causal modeling, the potential for (adversarial) misuse are all issues that may have adverse impacts when using our method. We do stress that these are not limitations inherent to our approach in particular, rather, these concerns arise for virtually any approach in a sensitive application. For an introductions to these issues, including a causal approach to them, see [4, 29].

Further, like any approach based on causality, our method relies on strong assumptions that are untestable in general, though they may be falsified in specific cases. Sequential ignorability in particular is a stronger assumption in cases with more mediating variables, or with a mediator that is causally influenced by many other variables (observed or unobserved). Such cases increase the number of opportunities for sequential ignorability to be violated for one of the mediators or by one of the many causes of a heavily influenced mediator.

Finally, intersectional fairness is not a purely statistical or algorithmic issue. As such, any technical method will require assumptions at least as strong as the causal assumptions we make. In particular, there are normative and subtle empirical issues embedded in any approach to fairness, such as the social construction of sensitive attributes, or the choice of which mediators may be considered resolving in our framework. For these reasons *we believe the burden of proof should fall on any approaches assuming the world (causal model) is already less unfair or that fairness interventions should be minimized*, for example by the use of resolving variables.

5 Related Work

Intersectionality. From the seeds of earlier work [13], including examples that motivated our experiments [14], intersectional feminism has developed into a rich interdisciplinary framework to analyze power and oppression in social relations [12, 43]. We refer especially to the work of Noble [37], and D'Ignazio and Klein [16], in the context of data and information technology. Other recent technical work in this area focuses on achieving guarantees across intersectional subgroups [20, 24, 27], including on computer vision tasks [7], or makes connections to privacy [18]. These do not take a causal approach or deal with ranking tasks. In our framework, intersectionality does not simply refer to a redefinition of multiple categorical sensitive attributes into a single product category or inclusion of interaction terms, as was done in recent work [20, 24, 27]. Specific problems may imply different constraints or interpretations for different sensitives attributes, as shown in the *moving company* example, where a mediator (e.g., weight-lifting ability) may be considered resolving for one sensitive attribute but not for another.

Causality and fairness. A growing literature on causal models for fair machine learning [10, 26, 30, 36, 55] emphasizes that fairness is a normative goal that relates to real world (causal) relationships. One contribution of the present work is to connect intersectionality and fair ranking tasks to this literature, and therefore to the rich literature on causal modeling. Some recent work in causal fairness focuses on the impact of learning optimal, fair policies, potentially under relaxations of standard causal assumptions that allow interference [28, 35]. Some of the most closely related work uses causal modeling to analyze intersectional fairness from a philosophical standpoint [6] or in a public health setting [22], but these are focused on foundations and interpretation, rather than on implementation or machine learning tasks.

Ranking and fairness. While the majority of the work on fairness in machine learning focuses on classification or risk prediction, there is also a growing body of work on fairness and diversity in ranking [2, 8, 9, 31, 45, 48, 49, 51, 52, 53], including a recent survey [54]. Yang et al. [49] consider intersectional concerns, although not in a causal framework. The authors observe that when representation constraints are stated on individual attributes, like race and gender, and when the goal is to maximize score-based utility subject to these constraints, then a particular kind of unfairness can arise, namely, utility loss may be imbalanced across intersectional groups. Barnabò et al. [3] study similar problem through explicitly modeling the trade-off between utility and fairness constraints. In our experiments we observed a small imbalance in utility loss across intersectional groups (1-5%) and will investigate the conditions under which this happens in future work. Finally, Wu et al. [48] apply causal modeling to fair ranking but estimates scores from observed ranks, uses causal discovery algorithms to learn an SCM, and does not consider intersectionality, while the present work considers the case when scores are observed and the SCM chosen *a priori.*

6 Conclusion

Our work builds on a growing literature for causal fairness to introduce a modeling framework for intersectionality and apply it to ranking. Experiments show that this approach can be flexibly applied to different scenarios, including ones with mediating variables, and the results compare reasonably to intuitive expectations we may have about intersectional fairness for those examples. The flexibility of our approach and its connection to causal methodology makes possible a great deal of future work including exploring robustness of rankings to unmeasured confounding [25] or uncertainty about the underlying causal model [41].

Future technical work can relax some assumptions under specific combinations of model structures, estimation methods, and learning task algorithms. For example, we have shown in experiments that the LTR task (without in-processing) with ListNet works reasonably well, but future work could identify the conditions when this insensitivity of a learned ranker to counterfactual transformations on the training data guarantees that counterfactual fairness will hold at test time, perhaps with explicit bounds on discrepancies due to issues like covariate shift. We proposed ranking on counterfactual scores, treating everyone as a member of the disadvantaged intersectional group, but there are other possible fair strategies. For any fixed baseline intersectional group, for example the most advantaged one, if we compute counterfactuals and treat everyone as though they belong to that fixed group, we would also achieve intersectional counterfactual fairness. The same is true if we treat everyone based on the average of their counterfactual values for all intersectional subgroups. Future work may explore whether any of these choices have formal or computational advantages, making them preferable in specific settings.

References

1 Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. *And it's biased against blacks. ProPublica*, 23, 2016.

2 Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *ACM SIGMOD*, pages 1259–1276, 2019. doi:10.1145/3299869.3300079.

3 Giorgio Barnabò, Carlos Castillo, Michael Mathioudakis, and Sergio Celis. Intersectional affirmative action policies for top-k candidates selection. *CoRR*, abs/2007.14775, 2020. arXiv:2007.14775.

4 Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *NIPS Tutorial*, 2017.

5 Emery Berger. CSRankings: Computer Science Rankings, 2017–2020. Online, retrieved June 2, 2020. URL: http://csrankings.org/.

6 Liam Kofi Bright, Daniel Malinsky, and Morgan Thompson. Causally interpreting intersectionality theory. *Philosophy of Science*, 83(1):60–81, 2016.

7 Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

8 L. Elisa Celis, Anay Mehrotra, and Nisheeth K. Vishnoi. Interventions for ranking in the presence of implicit bias. In *FAT**, pages 369–380. ACM, 2020. doi:10.1145/3351095.3372858.

9 L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, volume 107 of *LIPIcs*, pages 28:1–28:15, 2018. doi:10.4230/LIPIcs.ICALP.2018.28.

10 Silvia Chiappa. Path-specific counterfactual fairness. In *AAAI*, volume 33, pages 7801–7808, 2019.

11 Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Commun. ACM*, 63(5):82–89, 2020. doi:10.1145/3376898.

12 Patricia Hill Collins. *Black feminist thought: Knowledge, consciousness, and the politics of empowerment.* routledge, 2002.

13 Combahee River Collective. The Combahee river collective statement. *Home girls: A Black feminist anthology*, pages 264–74, 1983.

14 Kimberle Crenshaw. Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. *University of Chicago Legal Forum*, page 139, 1989.

15 Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE SP*, pages 598–617, 2016. doi:10.1109/SP.2016.42.

16 Catherine D'Ignazio and Lauren F Klein. *Data feminism.* MIT Press, 2020.

17 Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

18 James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. An intersectional definition of fairness. In *IEEE ICDE*, pages 1918–1921, 2020.

19 Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NIPS*, pages 3315–3323, 2016. URL: https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html.

20 Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *ICML*, pages 1939–1948, 2018.

21 Kosuke Imai, Luke Keele, and Teppei Yamamoto. Identification, inference and sensitivity analysis for causal mediation effects. *Statistical science*, pages 51–71, 2010.

22 John W Jackson and Tyler J VanderWeele. Intersectional decomposition analysis with differential exposure, effects, and construct. *Social Science & Medicine*, 226:254–259, 2019.

23 Pearl Judea. Causality: models, reasoning, and inference. *Cambridge University Press*, 2000.

24 Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, pages 2569–2577, 2018. URL: http://proceedings.mlr.press/v80/kearns18a.html.

25 Niki Kilbertus, Philip J. Ball, Matt J. Kusner, Adrian Weller, and Ricardo Silva. The sensitivity of counterfactual fairness to unmeasured confounding. In *UAI*, page 213, 2019. URL: http://proceedings.mlr.press/v115/kilbertus20a.html.

26 Niki Kilbertus, Mateo Rojas Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *NIPS*, pages 656–666, 2017.

27 Michael P. Kim, Amirata Ghorbani, and James Y. Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *AIES*, pages 247–254, 2019. doi:10.1145/3306618.3314287.

28 Matt Kusner, Chris Russell, Joshua Loftus, and Ricardo Silva. Making decisions that reduce discriminatory impacts. In *ICML*, pages 3591–3600, 2019.

29 Matt J Kusner and Joshua R Loftus. The long road to fairer algorithms. *Nature*, 2020.

30 Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *NIPS*, pages 4066–4076, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html.

31 Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. ifair: Learning individually fair data representations for algorithmic decision making. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 1334–1345. IEEE, 2019. doi:10.1109/ICDE.2019.00121.

32 Tie-Yan Liu. *Learning to Rank for Information Retrieval.* Springer, 2011. doi:10.1007/978-3-642-14267-3.

33 Joshua R Loftus, Chris Russell, Matt J Kusner, and Ricardo Silva. Causal reasoning for algorithmic fairness. *arXiv preprint*, 2018. arXiv:1805.05859.

34 Bhaskar Mitra, Nick Craswell, et al. *An introduction to neural information retrieval.* Now Foundations and Trends, 2018.

35 Razieh Nabi, Daniel Malinsky, and Ilya Shpitser. Learning optimal fair policies. In *International Conference on Machine Learning*, pages 4674–4682, 2019.

36 Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

37 Safiya Umoja Noble. *Algorithms of oppression: How search engines reinforce racism.* nyu Press, 2018.

38 Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 411–420, 2001.

39 James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512, 1986.

40 James M Robins. Semantics of causal dag models and the identification of direct and indirect effects. *Oxford Statistical Science Series*, pages 70–82, 2003.

41 Chris Russell, Matt J Kusner, Joshua Loftus, and Ricardo Silva. When worlds collide: integrating different counterfactual assumptions in fairness. In *NIPS*, pages 6414–6423, 2017.

42 Maya Sen and Omar Wasow. Race as a bundle of sticks: Designs that estimate effects of seemingly immutable characteristics. *Annual Review of Political Science*, 19, 2016.

43 Stephanie A Shields. Gender: An intersectionality perspective. *Sex roles*, 59(5-6):301–311, 2008.

44 Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search.* MIT press, 2000.

45 Julia Stoyanovich, Ke Yang, and H. V. Jagadish. Online set selection with fairness and diversity constraints. In *EDBT*, pages 241–252. OpenProceedings.org, 2018. doi:10.5441/002/edbt.2018.22.

46 Dustin Tingley, Teppei Yamamoto, Kentaro Hirose, Luke Keele, and Kosuke Imai. Mediation: R package for causal mediation analysis. *Journal of Statistical Software*, 2014.

47 Tyler VanderWeele. *Explanation in causal inference: methods for mediation and interaction.* Oxford University Press, 2015.

48 Yongkai Wu, Lu Zhang, and Xintao Wu. On discrimination discovery and removal in ranked data using causal graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2536–2544, 2018. doi:10.1145/3219819.3220087.

49 Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. Balanced ranking with diversity constraints. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6035–6042, 2019. doi:10.24963/ijcai.2019/836.

50 Ke Yang, Joshua R. Loftus, and Julia Stoyanovich. Causal intersectionality for fair ranking, 2020. arXiv:2006.08688.

51 Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *ACM SSDBM*, pages 22:1–22:6, 2017. doi:10.1145/3085504.3085526.

52 Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng, Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li, editors, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1569–1578. ACM, 2017. doi:10.1145/3132847.3132938.

53 Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2849–2855. ACM / IW3C2, 2020. doi:10.1145/3366424.3380048.

54 Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey, 2021. arXiv:2103.14000.

55 Junzhe Zhang and Elias Bareinboim. Fairness in decision-making – the causal explanation formula. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A Appendix

A.1 Proof of Proposition 2

Proposition 2 (Implementing counterfactually fair ranking) If the assumed causal model $\mathcal{M}$ is identifiable and correctly specified, implementations described above produce counterfactually fair rankings in the score based ranking and cf-LTR tasks.

Proof of Proposition 2. The proof is essentially by construction, but we provide more detail now for model $\mathcal{M}_1$. Fixing a baseline intersectional subgroup (g_0, r_0), the counterfactual training data in our implementation will use $Y_{(G,R)\leftarrow(g_0,r_0)}$, either by ranking these for score based ranking or training a predictive model for LTR. We wish to show that

$$\mathbb{P}(\hat{\boldsymbol{\tau}}(Y_{(G,R)\leftarrow(g,r)}) = k \mid X = x, (G,R) = (g,r)) \tag{5}$$

is unchanged under all counterfactual transformations, denoted by $Y_{(G,R)\leftarrow(g',r')}$, if the causal model has been correctly specified. First, we consider the case where the functions f_X, f_Y are known. If X is resolving, then

$$(Y_i)_{(G,R)\leftarrow(g_0,r_0)} = f_Y(x_i, g_0, r_0) + \epsilon_i^Y$$

for all i. In this case the conditional distribution of these scores (5) is invariant under counterfactual transformations $(g,r) \leftarrow (g',r')$ because x_i is held fixed, (g',r') will be substituted with the fixed baseline values (g_0, r_0), and the error term is exogenous and in particular its distribution does not change under transformations of (g,r). If X is not resolving then we use

$$(Y_i)_{(G,R)\leftarrow(g_0,r_0)} = f_Y(f_X(g_0,r_0) + \epsilon_i^X, g_0, r_0) + \epsilon_i^Y$$

Under counterfactual transformations $(g,r) \leftarrow (g',r')$ all of the inputs above stay fixed except for the error terms, and, as before, these errors do not depend on (g,r) so the training data scores have the desired distributional invariance. The semi-resolving case is similar.

For score based ranking $\hat{\boldsymbol{\tau}}$ sorts the counterfactual scores, denoted by $(Y_i)_{(G,R)\leftarrow(g_0,r_0)}$. Since the distributions of these scores are unchanged under counterfactual transformations as we just established, the probability for any score to equal a given rank k is also unchanged, hence $\hat{\boldsymbol{\tau}}$ is a counterfactually fair ranking. In cf-LTR, at test time the test data is first transformed to the intervened version $\mathbf{D}^{\text{test}}_{(G,R)\leftarrow(g_0,r_0)}$ before inputting to $\hat{\boldsymbol{\tau}}$. As before, the distribution of the predicted rank for observation i under any counterfactual transformation $(G,R) \leftarrow (g',r')$ is fixed to that of the distribution under $(G,R) \leftarrow (g_0,r_0)$, which depends only on the exogenous errors.

Finally, we relax the assumption that the functions f_X, f_Y are known. Since we have assumed the causal model is identifiable and correctly specified (in particular, it satisfies sequential ignorability in cases where the model has mediators), these functions can be estimated on the (training) data via any appropriate causal inference method. Hence, counterfactually fair ranking condition will hold approximately due to plug-in estimation error. ◀

A.2 Additional experimental results: score-based ranking

In this section, we show evaluation results of using our method on a more complicated data under a different causal model: a synthetic dataset with three sensitive attributes and one of them is a continuous or numeric attribute (e.g., age) under an assumed causal model $\mathcal{M}_5$ in Figure 2e.

Moving company + age is a variant of *moving company* dataset with 10, 000 records drawn from the causal model $\mathcal{M}_5$ in Figure 2e, with three sensitive attributes: gender G, race R, and age A, with edge weights $w(G \to X) = 0.95$, $w(R \to X) = 0$, $w(A \to X) = 0.05$, $w(G \to Y) = 0.1$, $w(R \to Y) = 0.1$, $w(A \to Y) = 0.1$, and $w(X \to Y) = 0.7$. Age A affects the weight-lifting ability score X and the qualification score Y in a piece-wise linear fashion, with X and Y decreasing for ages A above some thresholds. Specifically, the effect of age on X is negligible for ages below 45, then slightly negative, and more strongly negative above age 55. The mean age for White and Black individuals are 35 and 45 respectively. We use this dataset to showcase the applicability of our framework to cases with more than two sensitive attributes, and to cases where sensitive attributes may be continuous.

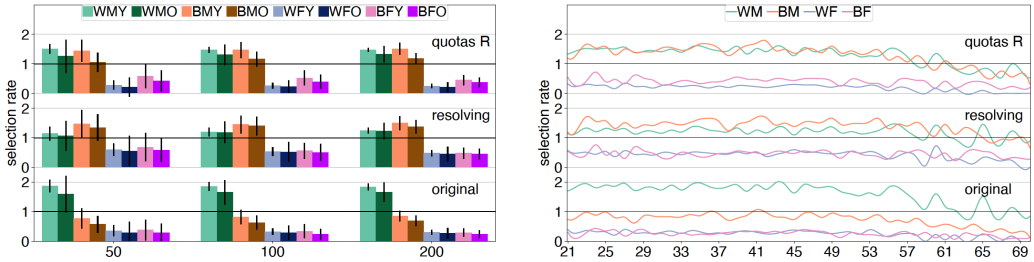
Figure 5 shows the performance of our methods in terms of demographic parity on *moving company+age* (over 100 executions), focusing on three different rankings: *original*, *resolving*, and *quotas R*. Recall that *moving company+age* includes a continuous sensitive attribute age in addition to gender and race. We present selection rates for two age groups, younger (age < 45) and older (age ≥ 45) in Figure 5a, and at each age in Figure 5b. We observe that in the *original* ranking, the bottom set of bars in Figure 5a, younger applicants are selected at a higher rate compared to older applicants within each intersectional group. For example,young White males and young Black males are both selected at higher rate than their older counterparts old White males and old Black males. Further, selection rates for racial and gender groups differ in the *original* ranking. For example, White males are selected at a much higher rate than other intersectional groups. These disparities in selection rates are preserved in the *quotas R* ranking, shown as the top set of bars in Figure 5a. Recall that the goal for *moving company+age* is to compute a ranking of the applicants that is free of racial and age discrimination while allowing for a difference in weight-lifting ability X between gender groups, thus treating X as resolving variable. In the counterfactually fair ranking *resolving*, the middle set of bars in Figure 5a, we observe an increase in selection rates for Black males, and also note that the age of the applicants does not materially affect their selection rates.

Figure 5b presents selection rates for each value of age, for each intersectional group on gender and race, at the top-200. Observe that the *original* ranking, shown in the bottom set of lines, exhibits a disparity in selection rates between the Black and the White applicants for all age values, and that selection rates drop substantially for all groups around age ≥ 50. The *quotas R* ranking, the top set of lines in Figure 5b, reduces the disparity in selection rates between racial groups (e.g., there is no gap between the lines for White males and Black males for any age), but it still shows a disparity by age, meaning that selection rates drop for all groups around age ≥ 50, just as they did in the original ranking. Finally, the counterfactually fair ranking *resolving*, shown as the middle set of lines in Figure 5b, reduces disparities in selection rates by both race and gender.

We also computed utility loss at top-k, based on the original Y scores (see Section 3.1 for details). For *moving company+age*, the loss of the counterfactually fair ranking *resolving* and of the *quotas R* ranking is at most 1% across the values of k.

A.3 Additional experimental results: rank-aware fairness measures

In this section, we report evaluation results of using a rank-aware fairness measure called **in-group fairness ratio (IGF-Ratio)** on *moving company*, *moving company+age*, and *COMPAS*. In-group fairness ratio (IGF-Ratio) is the simpler of two in-group fairness measures proposed in [49]. It captures an important intersectional concern that arises when an input ranking must be re-ordered (and thus suffer a utility loss) to satisfy some fairness or



(a)

(b) $k = 200$.

Figure 5 Demographic parity on the *moving company+age* dataset. The X-axis shows the top-k values of the rankings for (a) and shows the value of the attribute age A for (b). For both subplots, the Y-axis shows the selection rate, while each span of Y-axis represents different rankings and each color represents an intersectional group. The assumed causal model is $\mathcal{M}_5$ in Figure 2e. Figure 5a shows the results for the binarized attribute age A according to a threshold: younger ($\mathbf{Y}$): age < 45 and older ($\mathbf{O}$): age ≥ 45.

diversity constraint. Specifically, IGF-Ratio compares the amount of re-ordering within each intersectional groups, and considers a ranking fair if the corresponding loss is balanced across groups. Let us denote by $\boldsymbol{\tau}_{1\ldots k}$ the *set* of the top-k items in $\boldsymbol{\tau}$. For a given intersectional group g and position k, $\textit{IGF-Ratio}_k(\boldsymbol{\tau}, g)$ is the ratio of lowest score of any item from g in $\boldsymbol{\tau}_{1\ldots k}$ and the highest score of an item from g not in $\boldsymbol{\tau}_{1\ldots k}$. IGF-Ratio requires non-negative scores and ranges from $[0, 1]$, with higher values implying better in-group fairness. To make the scores non-negative, we increase the values of Y by $|\min(Y)|$.

Table 1 shows the results of in-group fairness ratio (IGF-Ratio) in counterfactually fair score-based ranking derived using our method on *moving company* (over 100 executions), *moving company+age* (over 100 executions), and *COMPAS*. To compute this measure, we cannot have any ties in the ranking. For *COMPAS*, we broke the ties by Y-score by randomly permuting the items within an equivalence class by score. Recall that IGF-Ratio ranges between 0 and 1 and that a higher value is better, since it indicates that the ratio of the score of the lowest-scoring selected item among the top-k and of the highest-scoring item not among the top-k is close to 1. Observe that most IGF-Ratio values are close to 1, meaning that there is only a limited amount of re-ordering of individuals within each intersectional group. Further, in-group fairness loss in terms of IGF-Ratio is balanced among intersectional groups in all cases, while some groups (e.g. White males) face a slightly lower but acceptable IGF-Ratio in the fair ranking *non-resolving*.

Table 1 IGF-Ratio on *moving company*, *moving company+age*, and *COMPAS*. A higher value is better: it indicates that the ratio of scores of the lowest-scoring selected item among the top-k and of the highest-scoring item not among the top-k is close to 1. In the table, $k_{1,2,3} = 50, 100, 200$ for *moving company* ($n = 2000$) and *moving company+age* ($n = 10,000$), and $k_{1,2,3} = 500, 1000, 1500$ for *COMPAS* ($n = 4162$). N/A is used when a particular intersectional group is not represented among the top-k.

Dataset	Ranking	k_1				k_2				k_3			
		WM	BM	WF	BF	WM	BM	WF	BF	WM	BM	WF	BF
moving company	non-res	0.98	0.94	0.93	0.94	0.96	0.95	0.91	0.92	0.94	0.94	0.89	0.89
	resolving	0.95	0.95	0.98	0.98	0.93	0.93	0.96	0.96	0.92	0.92	0.93	0.94
moving company+age	non-res	0.82	0.9	0.99	0.99	0.8	0.99	0.99	0.98	0.99	0.98	0.99	0.97
	resolving	0.99	0.99	0.99	0.94	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
COMPAS	non-res	N/A	1.00	N/A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	resolving	N/A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00