Lexicographically Fair Learning: Algorithms and Generalization

Emily Diana, Wesley Gill, Ira Globus-Harris, Michael Kearns, Aaron Roth, Saeed Sharifi-Malvajerdi

ediana@wharton.upenn.edu

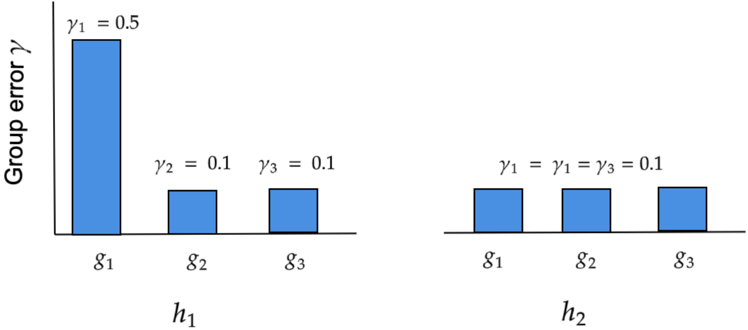
June 23, 2022

Motivation

We want our algorithms to treat different groups of people equitably.

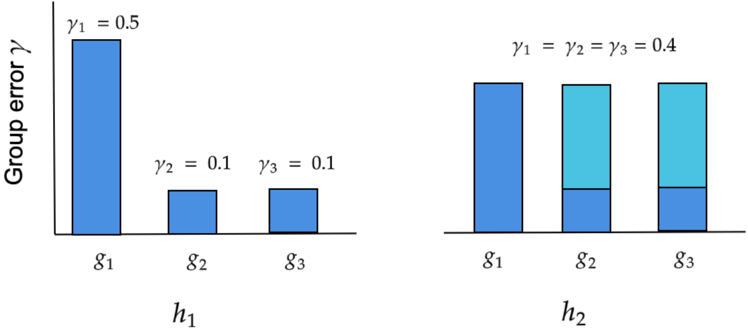
A Group Fairness Definition: Equality of Group Errors

"The algorithm should make the same number of mistakes on all groups."



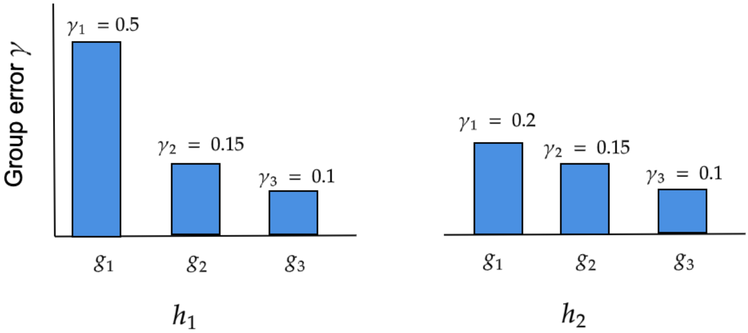
A Group Fairness Definition: Equality of Group Errors

"The algorithm should make the same number of mistakes on all groups."



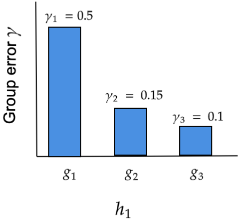
Alternative Group Fairness Definition: Minimax Group Fairness

"The number of errors made on the worst-off group should be minimized."



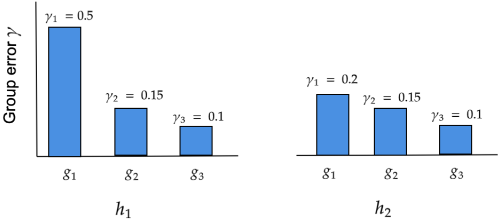
Extending Minimax Fairness: Lexicographic Fairness

"The number of errors made on the worst-off group should be minimized, and subject to that, the second-worst-off group's errors should be minimized, and subject to that..."



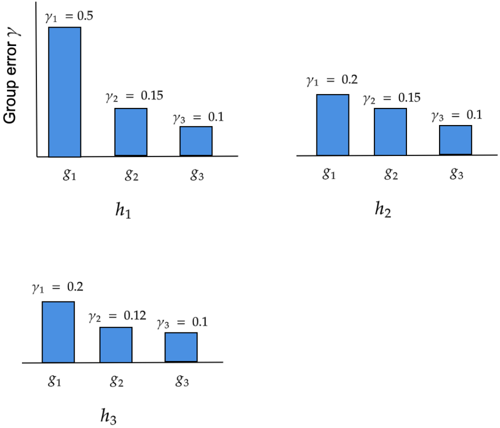
Extending Minimax Fairness: Lexicographic Fairness

"The number of errors made on the worst-off group should be minimized, and subject to that, the second-worst-off group's errors should be minimized, and subject to that. . . "

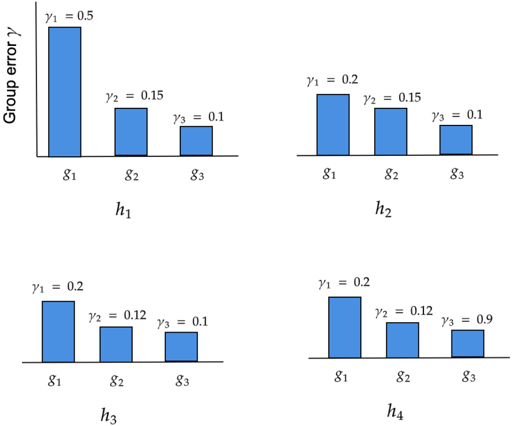


Extending Minimax Fairness: Lexicographic Fairness

"The number of errors made on the worst-off group should be minimized, and subject to that, the second-worst-off group's errors should be minimized, and subject to that..."

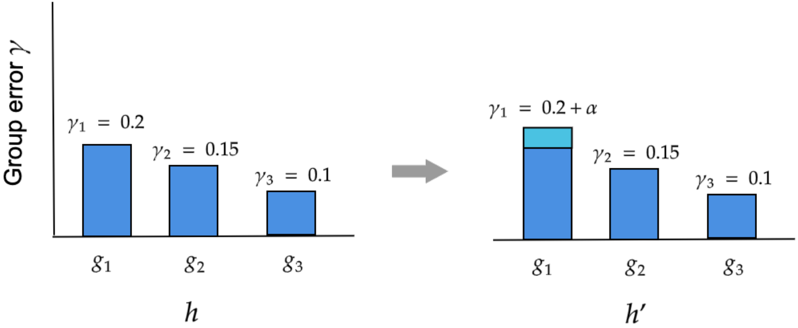


"The number of errors made on the worst-off group should be minimized, and subject to that, the second-worst-off group's errors should be minimized, and subject to that..."



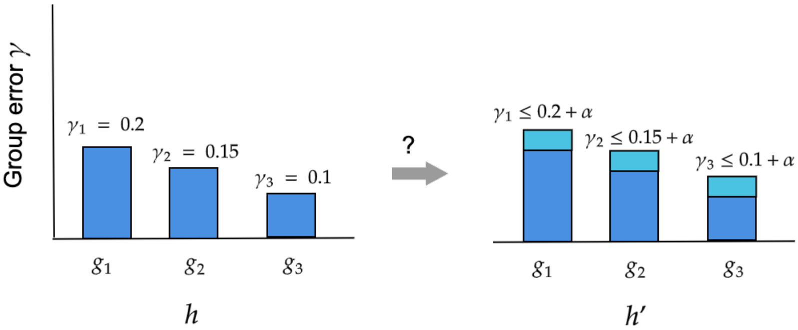
Approximating Lexicographic Fairness

We can only efficiently get approximate minmax-fair solutions.

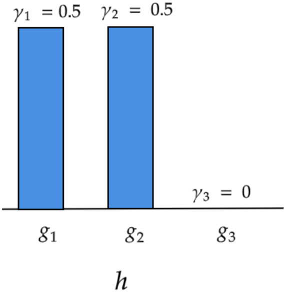


Approximating Lexicographic Fairness

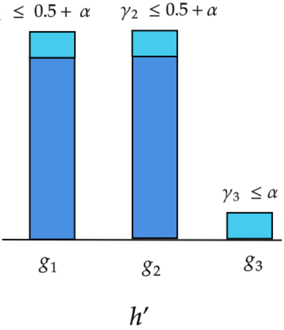
How do we generalize this to the lexifair setting?



Approximating Lexicographic Fairness

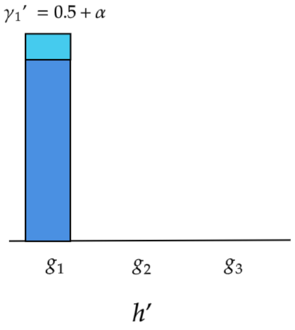


True lexifair solution

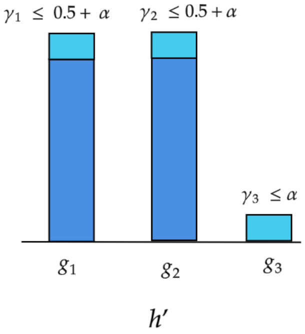


Approximate lexifair solution

Approximating Lexicographic Fairness

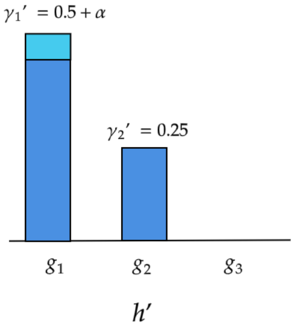


Approximate minimax solution

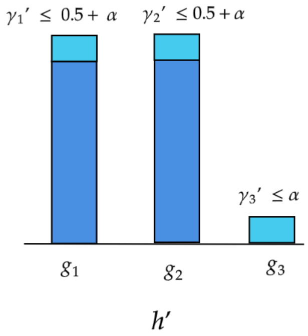


Approximate lexifair solution

Approximating Lexicographic Fairness

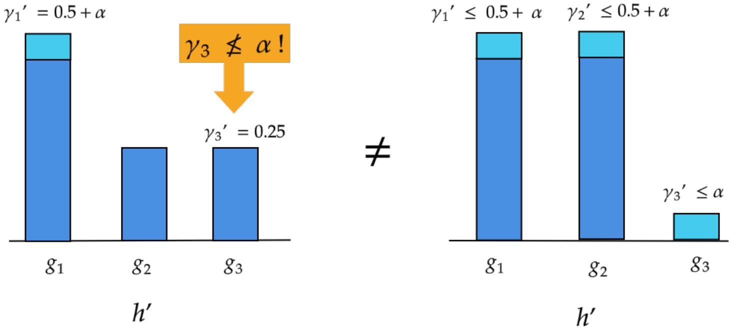


Approximate min of top 2 group errors



Approximate lexifair solution

Approximating Lexicographic Fairness



Approximate min of top 3 group errors

Approximate lexifair solution

Approximate Lexicographic Fairness: A Stable Definition

Definition (Approximate Lexicographic Fairness)

Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. Let $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$, and define

$$\mathcal{H}^{\vec{\epsilon}}_{(0)} \triangleq \text{ the entire model class } \mathcal{H},$$

$$\mathcal{H}^{\vec{\epsilon}}_{(j)} \triangleq \text{models in } H^{\vec{\epsilon}}_{j-1} \text{ that have the smallest } j\text{th group error rate up to an } \epsilon_j \text{ approximation.}$$

A model h satisfies (ℓ, α)-lexicographic fairness ("lexifairness") if $h \in \mathcal{H}^{\vec{\epsilon}}_{(\ell)}$ for some $\vec{\epsilon}$ that is component-wise less than α.

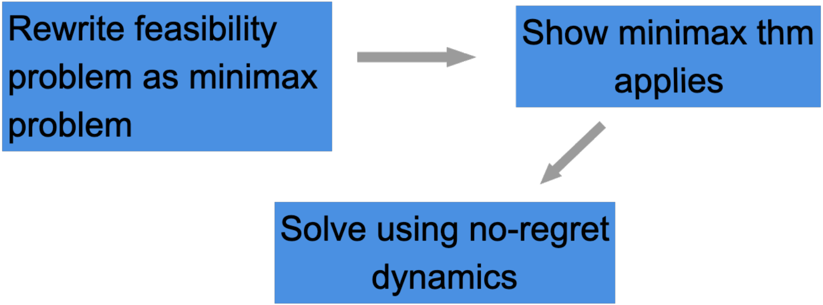
Convex Lexifairness

- 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 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.
- We get around this by replacing constraints on the k'th highest error groups with constraints on the *sums* of all k-tuples of group errors.

Our results

- Define a stable and convex version of approximate lexifairness.
- Derive oracle-efficient algorithms for finding approximately lexifair solutions.
- Show that when the underlying empirical risk minimization problem absent fairness constraints is convex, our algorithms are provably efficient.
- Show that approximate lexifairness generalizes: approximate lexifairness on the training sample implies approximate lexifairness on the true distribution w.h.p.

Oracle-efficient algorithms to achieve approximate lexifairness



- In regression setting, learner plays Online Projected Gradient Descent.
- In classification setting, learner plays Follow-the-Perturbed-Leader.

Algorithmic Formulation

- Our approach to find lexifair models is to **recursively** find the minimax (over sums of group error rates) rates
- Our algorithms return a model achieving those minimax rates, and hence that model will be lexifair.
- At level j, in an inductive fashion, we are given the minimax rates $\eta_1, \ldots, \eta_{j-1}$ from previous rounds, and we want to estimate η_j
- Can then dictate that every sum of j group error rates is at most η_j
- Writing the Lagrangian of this linear program

Let $L_{i_r}(h)$ indicate the loss incurred by the model h on the i_r'th group. Then the Lagrangian for this linear program can be written as

$$\mathcal{L}_j\left((h, \eta_j), \lambda\right) = \eta_j + \sum_{r=1}^{j} \sum_{\{i_1, \ldots, i_r\} \subseteq [K]} \lambda_{\{i_1, i_2, \ldots, i_r\}} \cdot \left(L_{i_1}(h) + \ldots + L_{i_r}(h) - \eta_r\right) \tag{1}$$

Algorithmic Formulation: Two Player Zero-Sum Game

Can find a minimax solution for this Lagrangian with a zero-sum game between a (L)earner and a (A)uditor:

- At each round t, there is a weighting over groups determined by A
- L (best) responds by computing model h_t to minimize the weighted prediction error
- A updates group weights using online projected gradient descent with respect to group errors achieved by h_t
- L's final model M is uniform distribution over all of h_t's produced

Finding Lexifair Regression Model

ALGORITHM 1: `LexiFairReg`: Finding a Lexifair Regression Model

Input: $S = \cup_{k=1}^{K} G_k$ data set consisting of K groups, (ℓ, α) desired fairness parameters, loss function parameters L_M

for $j = 1, 2, \ldots, \ell$ **do**

  Set $T_j = O(\frac{1}{\alpha^2})$;

  $(\hat{\theta}_j, \hat{\eta}_j) = \texttt{RegNR}(T_j; \hat{\eta}_1, \ldots, \hat{\eta}_{j-1})$ (Calling Algorithm 2)

Output: (ℓ, α)-convex lexifair model $\hat{\theta}_\ell$

- At each level j, we employ a subroutine in which the Learner plays Online Projected Gradient Descent and the Auditor best responds

Two Player Game Subroutine

ALGORITHM 2: `RegNR`: jth round

Input: Number of rounds T, previous estimates $(\eta_1, \ldots, \eta_{j-1})$

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 (θ^t, η_j^t);

Auditor best responds: $\lambda^t = \lambda_{\text{best}}(\theta^t, \eta_j^t; (\eta_1, \ldots, \eta_{j-1}))$;

Learner updates its actions using Projected Gradient Descent:

$$\theta^{t+1} = \text{Proj}_{\Theta}\left(\theta^t - \eta \cdot \nabla_\theta \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$$

$$\eta_j^{t+1} = \text{Proj}_{[0, j \cdot L_M]}\left(\eta_j^t - \eta' \cdot \nabla_{\eta_j} \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$$

Output: the average play $\hat{\theta} = \frac{1}{T}\sum_{t=1}^{T} \theta^t \in \Theta$, and $\hat{\eta}_j = \frac{1}{T}\sum_{t=1}^{T} \eta_j^t \in [0, j \cdot L_M]$.

Algorithm Overview: Auditor's Best Response

Auditor plays maximum weight on most violated constraint:

ALGORITHM 3: The Auditor's Best Response (λ_{best}): jth round

Input: Learner's play (h, η_j), previous estimates $(\eta_1, \ldots, \eta_{j-1})$
Compute $L_k(h)$ for all groups $k \in [K]$;
Find the top j elements of vector $(L_1(h), \ldots, L_K(h))$ and call them:
$L_{\bar{h}(1)}(h) \geq \ldots \geq L_{\bar{h}(j)}(h)$;
if $\forall r \leq j$: $L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) \leq \eta_r$ **then** $\lambda_{out} = 0$;
else Let $r^\star \in \text{argmax}_{r \leq j} \left(L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) - \eta_r \right)$, $\lambda_{out} = \lambda^\star$;
Output: $\lambda_{out} \in \Lambda_j$

Generalization

- Our ability to prove out of sample bounds crucially relies on our definitional choices that ensure stability.
- Specifically, we show that if:
  1. Our base class $\mathcal{H}$ satisfies a standard uniform convergence bound across every group:
     For distribution $\mathcal{P}$ and $\delta > 0$ there exists $\beta(\delta)$ such that

$$\Pr_S \left[\max_{h \in \mathcal{H}, k \in [K]} \left| L_k\left(h, S\right) - L_k\left(h, \mathcal{P}\right) \right| > \beta(\delta) \right] < \delta$$

  2. We have a model that is approximately convex lexifair on our dataset $S \sim \mathcal{P}^n$

  then our model is also appropriately convex lexifair on the underlying distribution.

Generalization for Convex Lexifairness

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 $\alpha' = \alpha + 2\ell\beta(\delta)$.

Generalization for Convex Lexifairness: Classification

Note that in the case of classification with 0/1 loss, the sample complexity is *polynomial* in the relevant parameters ℓ, α and VC dim.

Suppose $\mathcal{H}$ is a class of binary classifiers with VC dimension $d_{\mathcal{H}}$ For every $\mathcal{P}$, every data set $S \equiv \{G_k\}_k$ of size n sampled $i.i.d.$ from $\mathcal{P}$, if a randomized 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

$$\min_{1 \le k \le K} |G_k| = \Omega\left(\frac{l^2\left(d_{\mathcal{H}} \log(n) + \log(K/\delta)\right)}{\alpha^2}\right)$$