Reliable and Trustworthy Artificial Intelligence

Lecture 12: Group Fairness

Martin Vechev, Nikola Jovanovic

ETH Zurich

Fall 2022

Recap: Fairness Definitions

Individual Fairness

Similar individuals should be treated similarly.

(generally, a deterministic specification)

Group Fairness

On average, different groups are treated similarly.

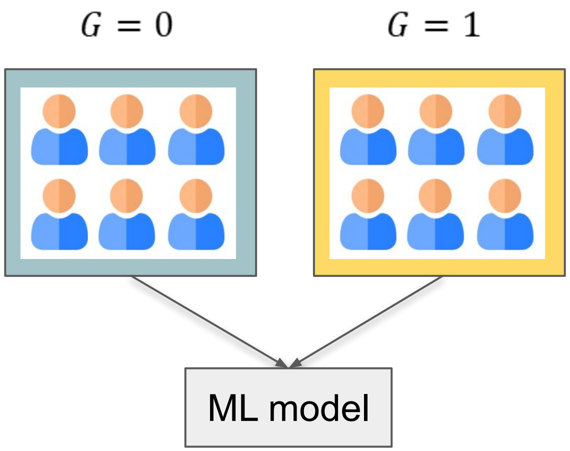
(generally, a probabilistic specification)

Counterfactual Fairness

Protected characteristics should not affect decisions causally.

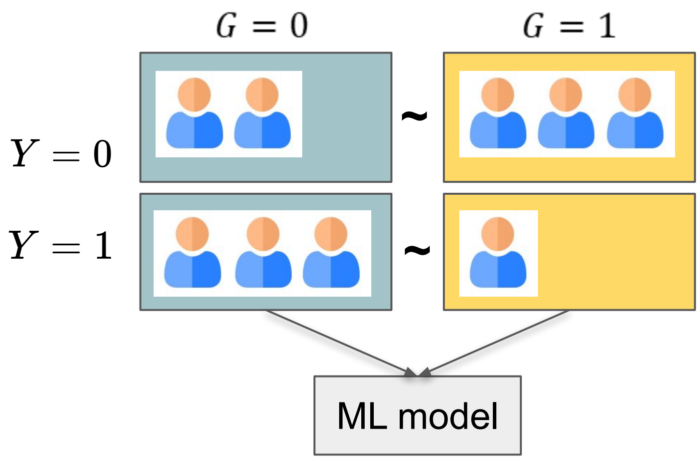
Recap: Group Fairness Constraints

Demographic parity



$$\mathbb{P}(\hat{Y} = 1 | G = 0) = \mathbb{P}(\hat{Y} = 1 | G = 1)$$

Equalized odds



$$\mathbb{P}(\hat{Y} = 1 | Y = 0, G = 0) = \mathbb{P}(\hat{Y} = 1 | Y = 0, G = 1)$$
$$\mathbb{P}(\hat{Y} = 1 | Y = 1, G = 0) = \mathbb{P}(\hat{Y} = 1 | Y = 1, G = 1)$$

How do we train models that satisfy group fairness?

Three Classes of Techniques

Pre-processing

Transform the data into de-biased representations, such that any classifier trained on them is fair

In-training

Modify the training of the model to incorporate a fairness constraint, making the resulting model more fair

Post-processing

Adjust the predictions of pre-trained models to make them less unfair

Post-processing Methods

Adjust the predictions of pre-trained models to make them less unfair

- 🟢 Works for **any** black-box classifier
- 🟢 Efficient; does not require training new models
  - Retraining sometimes expensive/impossible
- 🔴 May lack flexibility for a good fairness/accuracy tradeoff
- 🔴 Requires test-time access to sensitive attributes

Post-processing Methods

Adjust the predictions of pre-trained models to make them less unfair

Example (Hardt et al., ‘16):

- Given a binary classifier g, where $\mathrm{g(x)}$ is the output probability
  - Standard setting: $\mathrm{g(x)} > 0.5 \rightarrow$ favorable (e.g., loan granted)
  - However, this prediction may be unfair to some groups
- Instead, calculate different classification thresholds $\{\mathrm{t_0}, \mathrm{t_1}\}$ for two sensitive groups $\mathrm{s}=0$ and $\mathrm{s}=1$, based on the desired tradeoff
- Then:
  - if $\mathrm{s} = 0$, predict favorable outcome if $\mathrm{g(x)} > \mathrm{t_0}$
  - if $\mathrm{s} = 1$, predict favorable outcome if $\mathrm{g(x)} > \mathrm{t_1}$

In-training Methods

Modify the training of the model to incorporate a fairness constraint

🟢 Highest potential for a good tradeoff as we can focus on a particular model

🟢 No need to know the sensitive attribute at test time (unlike post-processing), but does need it at training time

🔴 Needs access to the training pipeline

🔴 No generality; specialized solutions for a particular task / model class

In-training Methods

Modify the training of the model to incorporate a fairness constraint

Example (Zafar et al., '17):

- Add soft fairness constraints to loss minimization
- Relax constraints to make optimization feasible

standard classification **loss**

$$\begin{array}{ll} \text{minimize} & \mathcal{L}_{clf}(\Theta) \\ \text{subject to} & P(\hat{y}=1|s=0) - P(\hat{y}=1|s=1) \leq \epsilon \\ & P(\hat{y}=1|s=0) - P(\hat{y}=1|s=1) \geq -\epsilon \end{array}$$

constraints for (approximate) demographic parity

Pre-processing Methods (Fair Representation Learning)

Transform data x into de-biased representations z s.t. <u>any</u> classifier trained on z is fair

Property of the transformation: post-processing cannot increase dependence on the sensitive attribute – known as the *data processing inequality* in information theory

🟢 Agnostic to later steps; z can be used for any downstream task / model class

Efficient, flexible/transferable, does not need trust towards downstream users

🟢 Downstream classifier does not need to know the sensitive attribute (at neither train nor test time)

🔴 May overly sacrifice accuracy for fairness as it is unaware of downstream task/classifier

🔴 The learned representation does not protect against adversarial downstream parties

We will look at three FRL examples in the rest of the lecture

Fair Representation Learning (FRL): Notation

Data $(x, s) \in R^d \times \{0, 1\}$, sampled from a joint distribution X

Encoder f: $R^d \times \{0, 1\} \rightarrow R^{d'}$ creates representations $z = f(x, s)$, induces joint distribution Z on (z, s)

Classifier g: $R^{d'} \rightarrow \{0, 1\}$, trained for a binary prediction task ($z \rightarrow y$)

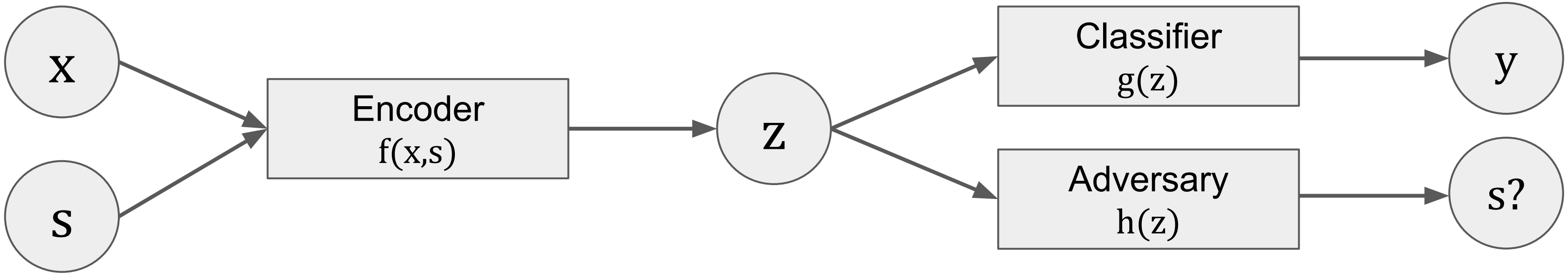
Adversary h: $R^{d'} \rightarrow \{0, 1\}$, aims to guess s from representations ($z \rightarrow s$)
The adversary concept will be used to reason about fairness properties of z

We further define some shorthands

- Z_0 and Z_1 are the conditional distributions of z where $s=0$ and $s=1$, respectively
- $p_0(z) := P(z \mid s = 0)$ and similarly $p_1(z) := P(z \mid s = 1)$ are the densities

LAFTR (Madras et al., '18)

Jointly trains the encoder f, classifier g, and adversary h (modeled as NNs)



$$\min_{f,g} \max_{h \in \mathcal{H}} \left(\mathcal{L}_{clf}(f(x,s),g) - \gamma \cdot \mathcal{L}_{adv}(f(x,s),h) \right)$$

Goal: learn representations z that are predictive of y but not predictive of s

Bounding Unfairness with the Optimal Adversary

We can use the concept of the **adversary** to upper bound the **unfairness** of **any** downstream classifier!

Ex: recall the definition of the **demographic parity (DP)** group-fairness constraint

$$P(g(z) = 1|s = 0) = P(g(z) = 1|s = 1)$$

We can turn this hard constraint into a soft unfairness measure of g, **DP-distance**

$$\Delta^{DP}_{\mathcal{Z}_0,\mathcal{Z}_1}(g) := \left| \mathop{\mathbb{E}}_{z\sim\mathcal{Z}_0} g(z) - \mathop{\mathbb{E}}_{z\sim\mathcal{Z}_1} g(z) \right|$$

distance 0 means g satisfies demographic parity, perfect fairness

distance 1 means g is maximally unfair towards one sensitive group

Bounding Unfairness with the Optimal <u>Adversary</u>

Let us also define the **balanced accuracy** of an adversary h

$$BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h) := \frac{1}{2}\left(\mathop{\mathbb{E}}_{z\sim\mathcal{Z}_0}(1-h(z)) + \mathop{\mathbb{E}}_{z\sim\mathcal{Z}_1} h(z)\right)$$
$$= \frac{1}{2}\int_{\mathcal{Z}}\Big(p_0(z)(1-h(z)) + p_1(z)h(z)\Big)dz$$

BA is a group-normalized accuracy (useful for imbalanced datasets)

- Has values in the interval $[0.5, 1]$
- If h always predicts sensitive group $1 \rightarrow$ balanced accuracy 0.5
- As we will see later, depending on p_0 and p_1, one can get balanced accuracy 1

Intuition: as $h(z)$ is either 0 or 1, for each z, adversary chooses between two groups by deciding what behavior h should have, and "selects" $p_0(z)$ or $p_1(z)$

Bounding Unfairness with the Optimal Adversary

$$BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h) = \frac{1}{2}\int_{\mathcal{Z}} \Big(p_0(z)(1-h(z)) + p_1(z)h(z)\Big)dz$$

What is the **optimal** adversary?

$$h^\star(z) = \mathbb{1}\{p_1(z) \geq p_0(z)\}$$

Predicts the group where the likelihood of z under the corresponding conditional distribution (Z_0 or Z_1) is greater. Note that the worst case for adversary is when **distributions are equal**. Then it is **impossible** to get balanced accuracy above 0.5

Generally intractable for NN encoders (as we cannot exactly compute the two densities)

Bounding Unfairness with the Optimal Adversary

Key result: DP-distance (unfairness) of **any** downstream classifier g trained on representations z is upper bounded by the balanced accuracy of the **optimal adversary** on representations z (Madras et al., ‘18):

$$\Delta^{DP}_{\mathcal{Z}_0,\mathcal{Z}_1}(g) \leq 2 \cdot BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h^{\star}) - 1$$

(*Proof in this week’s exercise)*

How can we use this?

LAFTR: Theoretically-principled FRL

$$\min_{f,g} \max_{h \in \mathcal{H}} \left(\mathcal{L}_{clf}(f(x,s),g) - \gamma \cdot \mathcal{L}_{adv}(f(x,s),h) \right)$$

LAFTR approximates the optimal adversary h^* via some adversary h used in training

🟢 This heuristic can sometimes lead to empirically good fairness

🔴 Hard non-convex min-max optimization $\to$ usually not solved optimally

🔴 Assumes the optimal adversary is in some family H (e.g., 2x100 NNs)

=> There can be stronger adversaries than h (with higher balanced accuracy)

We show experimental results on that later when discussing FNF

End-to-end fairness is overestimated (it really may be much less)

Can we produce <u>provably fair</u> representations?

Background: Normalizing Flows (Rezende & Mohamed, '15)

Generative models that transform a known distribution q into a learned distribution p

Key steps (for a trained flow):

1. Sample x from a known distribution q (with known density q(x), e.g., Gaussian)
2. Apply an **invertible** function z = f(x) (flow architecture ensures invertibility)
3. Use **change of variables** to compute the density of the new distribution at z (not possible for e.g., VAEs or GANs):

$$\log p(z) = \log q(f^{-1}(z)) + \log \left| det \frac{\partial f^{-1}(z)}{\partial z} \right|$$

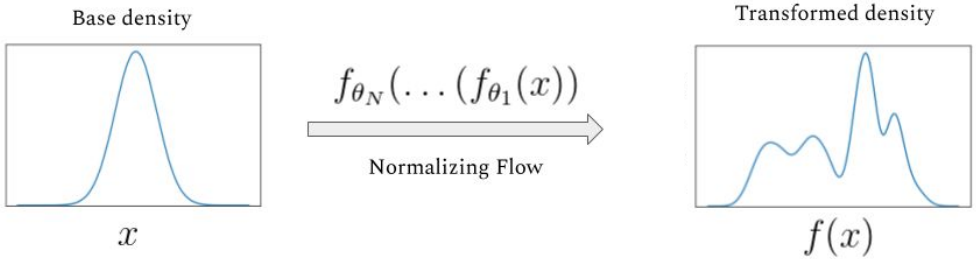


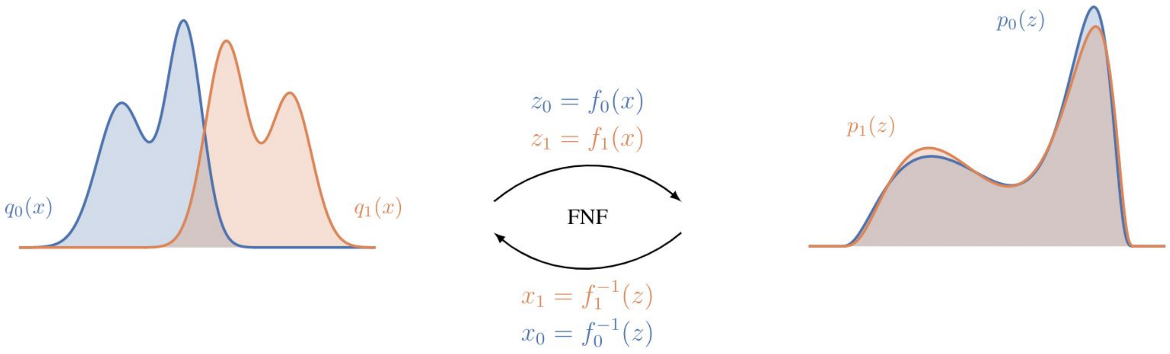
figure credit: Simon Boehm

Fair Normalizing Flows - FNF (Balunovic et al., ‘22)

Key idea: learn two normalizing flows f_0 and f_1 as encoders for Z_0 and Z_1, respectively

If we know the densities of the original data (conditioned on the sensitive attribute) $q_0(x)$ and $q_1(x)$, the normalizing flows allow us to get $p_0(z)$ and $p_1(z)$

To get an **estimate** of the densities $q_0(x)$ and $q_1(x)$ of the original data, one can use popular density estimation methods (e.g., Gaussian Mixture Model)



FNF: Provable Unfairness Upper Bound

We will compute an **upper bound T on DP-distance** using the inequality from before:

$$\Delta^{DP}_{\mathcal{Z}_0,\mathcal{Z}_1}(g) \leq 2 \cdot BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h^\star) - 1 \boxed{\leq T \text{ (w.p. 1-}\varepsilon)}$$

Recall:
$h^\star(z) = \mathbb{1}\{p_1(z) \geq p_0(z)\}$

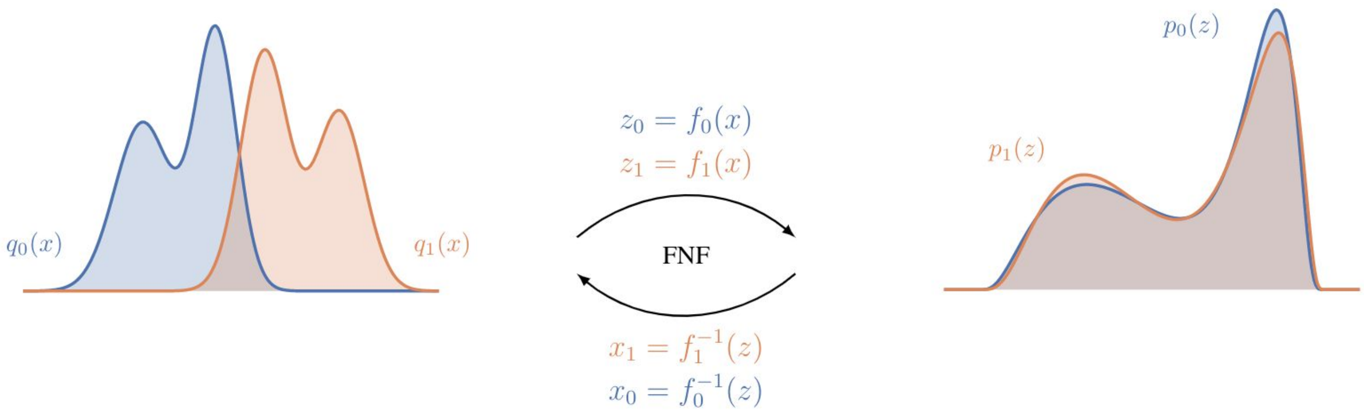
1. Start from n data samples $\{x_1, \ldots, x_n\}$ (the given dataset)
2. For each sample x, compute $q_0(x)$ (or $q_1(x)$ if $s=1$) using the previously fitted density estimation model
3. Apply the encoder to get $z=f_0(x)$, and use the flows to get $p_0(z)$ and $p_1(z)$
4. Use $p_0(z)$ and $p_1(z)$ to estimate the optimal adversary h^* and then **upper bound its balanced accuracy *BA*** with probability $1\text{-}\varepsilon$ (Hoeffding's inequality)
5. Use the inequality of (Madras et al., '18) above to **upper bound the DP-distance** of **any** downstream classifier trained on representations z with high probability

FNF: Training Normalizing Flows

We are not done yet – without a training procedure that enforces fairness, our proof produces an upper bound that is **sound** but **loose** or even vacuous

To get tight bounds: train the flows to promote low accuracy of h^*

In other words: **minimize the distance of distributions** Z_0 **and** Z_1



FNF: Training Normalizing Flows to Minimize KL Divergence

Minimize **symmetrized KL divergence** between Z_0 and Z_1

Combine with standard classification loss, with tradeoff parameter γ

Note: we include a classifier g in training to maintain utility of representations – this **does not affect the guarantees**

Algorithm 1 Learning Fair Normalizing Flows

Input: N, B, γ, q_0, q_1
Initialize g, f_0, f_1 with parameters $\theta_g, \theta_0, \theta_1$
for $i = 1$ **to** N **do**
  for $j = 1$ **to** B **do**
    Sample $\boldsymbol{x}_0^j \sim q_0, \boldsymbol{x}_1^j \sim q_1$
    $\boldsymbol{z}_0^j = f_0(\boldsymbol{x}_0^j)$
    $\boldsymbol{z}_1^j = f_1(\boldsymbol{x}_1^j)$
  end for
  $\mathcal{L}_0 = \frac{1}{B}\sum_{j=1}^{B}(\log p_0(\boldsymbol{z}_0^j) - \log p_1(\boldsymbol{z}_0^j))$
  $\mathcal{L}_1 = \frac{1}{B}\sum_{j=1}^{B}(\log p_1(\boldsymbol{z}_1^j) - \log p_0(\boldsymbol{z}_1^j))$
  $\mathcal{L} = \gamma(\mathcal{L}_0 + \mathcal{L}_1) + (1-\gamma)\mathcal{L}_{clf}$
  Update $\theta_s \leftarrow \theta_s - \alpha\nabla_{\theta_s}\mathcal{L}$, for $s \in \{0, 1\}$
  Update $\theta_g \leftarrow \theta_g - \alpha\nabla_{\theta_g}\mathcal{L}$
end for

FNF: How Tight is the Provable Upper Bound?

Recall: for some methods like LAFTR we can find adversaries from a different model class H (than the ones used in training) with **much higher (balanced) accuracy,** which implies **higher unfairness** of representations than estimated by the method

FNF is the 1st method to offer a **tight provable upper bound** (with a minor accuracy drop)

	Acc of g	Adv BA		Max Adv BA
		$h \in \mathcal{H}$	$h \notin \mathcal{H}$	
ADV FORGETTING (Jaiswal et al., 2020)	85.99	66.68	74.50	✗
MAXENT-ARL (Roy & Boddeti, 2019)	85.90	50.00	85.18	✗
LAFTR (Madras et al., 2018)	**86.09**	72.05	84.58	✗
FNF (our work)	84.43	N/A	**59.56**	**61.12**

tight

downstream classifier

empirical adversaries

provable upper bound

FNF: Summary

- **Provable upper bound** on unfairness of **any** downstream classifier
- Efficient training that reduces adversary's success => low empirical unfairness
- The guarantees only hold for **estimated** densities $q_0(x)$ and $q_1(x)$ (not real ones)

=> Guarantees technically do not hold in practice, they only hold when:

1) We can provably bound the distance between estimated and real densities
2) The data distribution is known

In most realistic use-cases, neither of these holds – this is a **major limitation**

Can we produce <u>provably fair</u> representations <u>with no restrictive assumptions</u>?

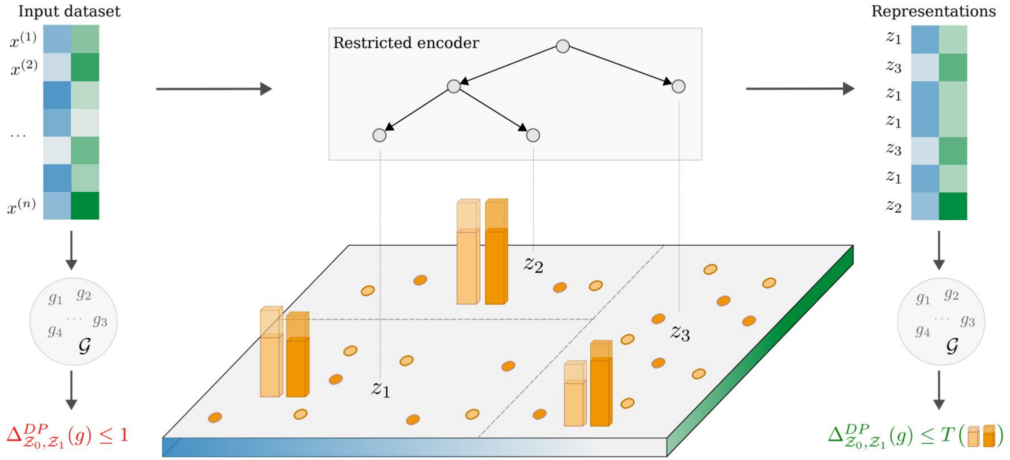
Fairness with Restricted Encoders - FARE (Jovanovic et al., '22)

Key idea: **restrict the space of representations** to a finite set

FNF: estimate $q_0(x)$ → get $p_0(z)$ → upper bound $BA(h^*)$

FARE: **directly upper bound $p_0(z)$** → upper bound $BA(h^*)$

As the space of z is finite we can do this tightly from given samples



FARE: Restricted Encoders

f: $R^d \times \{0, 1\} \rightarrow \{z_1, \dots, z_k\}$ that map each x to one of k possible representations z_i (*cells*)

Transform and upper bound the balanced accuracy (to get an unfairness upper bound):

$$BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h^*) := \frac{1}{2}\left(\mathop{\mathbb{E}}_{z\sim\mathcal{Z}_0}(1-h^*(z)) + \mathop{\mathbb{E}}_{z\sim\mathcal{Z}_1} h^*(z)\right)$$

Expectation of a discrete RV

$$= \frac{1}{2}\left(\sum_{i=1}^{k} p_0(z_i)\cdot(1-h^*(z_i)) + \sum_{i=1}^{k} p_1(z_i)\cdot h^*(z_i)\right)$$

Definition of optimal adversary

$$= \frac{1}{2}\left(\sum_{i=1}^{k} \max\left(p_0(z_i), p_1(z_i)\right)\right)$$

2x Bayes' rule

$$= \underbrace{\sum_{i=1}^{k} \underbrace{P(z=z_i)}_{\text{cell prior}} \cdot \underbrace{\max\left(\underbrace{(1/2P(s=0))}_{\alpha_0\ \text{(base rate)}} \cdot P(s=0|z=z_i), \underbrace{(1/2P(s=1))}_{\alpha_1\ \text{(base rate)}} \cdot P(s=1|z=z_i)\right)}_{c_i\ \text{(per-cell balanced accuracy)}}}_{\text{prior-weighted per-cell balanced accuracy}}$$

Bayes' rule that we use:

$$p_0(z_i) := P(z=z_i|s=0) = \frac{P(s=0|z=z_i)P(z=z_i)}{P(s=0)}$$

FARE: Provable Upper Bound

$$BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h^*) = \underbrace{\sum_{i=1}^{k} \underbrace{P(z = z_i)}_{\text{cell prior}} \cdot \underbrace{\max\left(\underbrace{(1/2P(s=0))}_{\alpha_0 \text{ (base rate)}} \cdot P(s=0|z=z_i), \underbrace{(1/2P(s=1))}_{\alpha_1 \text{ (base rate)}} \cdot P(s=1|z=z_i) \right)}_{c_i \text{ (per-cell balanced accuracy)}}}_{\text{prior-weighted per-cell balanced accuracy}}$$

We can upper bound this expression in three steps with a finite dataset, using Clopper-Pearson binomial CI (Steps 1, 2) and Hoeffding's inequality (Step 3):

(Step 1) Bounding base rates: $\alpha_0 < u_0$ and $\alpha_1 < u_1$ (with error ε_b) using the training set

(Step 2) Bounding per-cell balanced accuracy: $c_i \leq t_i$ (with error ε_c) using the validation set

(Step 3) Bounding the final sum: $\sum P(z = z_i) \cdot t_i \leq S$ (with error ε_s) using the test set

Union bound: total error is $\varepsilon = \varepsilon_b + \varepsilon_c + \varepsilon_s$

$$\Delta^{DP}_{\mathcal{Z}_0,\mathcal{Z}_1}(g) \leq 2 \cdot BA_{\mathcal{Z}_0,\mathcal{Z}_1}(h^\star) - 1 \leq 2S - 1 = T \text{ (w.p. 1-}\varepsilon)$$

FARE: Training Restricted Encoders

The upper bound holds for **any** restricted encoder

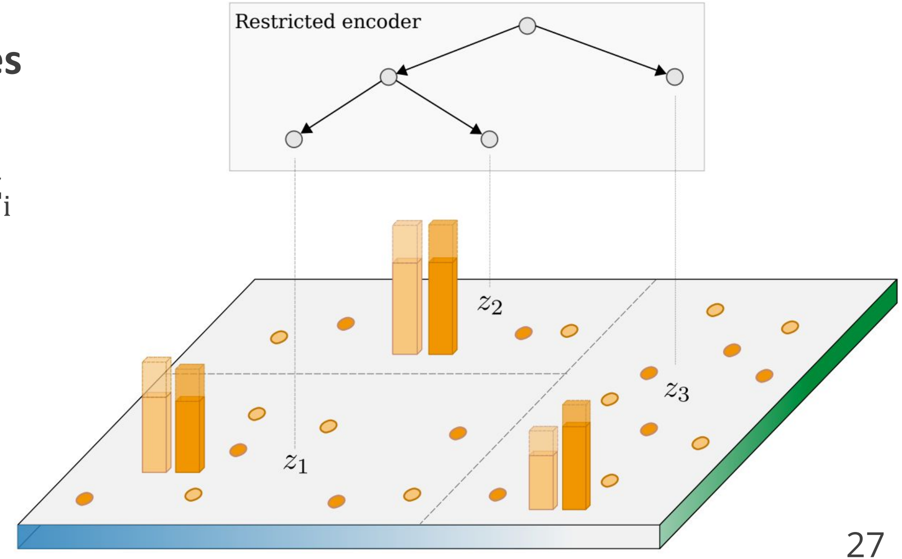
However, as before, the bound is useful only if in practice we can train restricted encoders that allow for **good empirical fairness/accuracy tradeoffs** and **tight bounds**

- Possible issue: expressivity of the representation space

One instantiation: **fairness-aware decision trees**

All datapoints in the same leaf are mapped to z_i (median of all such datapoints)

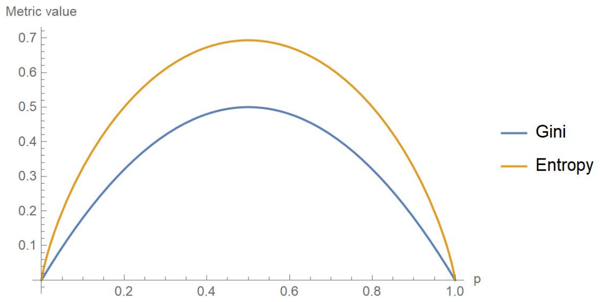
🟢 Discreteness by design, explicit control of proof-influencing parameters (e.g., #cells)



Recap: Decision Trees

Standard tree construction procedure used for binary classification tasks $x \rightarrow y$:

- Start from the full training set D_{root} of examples in the root node
- In each node, the current set D is **split** according to feature j and threshold v into
  $D_L = \{(x, y) \in D \mid x_j \leq v\}$ (*left child*) and $D_R = D \setminus D_L$ (*right child*)
  to minimize a criterion such as **Gini impurity** (weighted by $|D_L|$ and $|D_R|$):
  $$Gini_y(D) = 2p_y(1 - p_y) \in [0, 0.5]$$
  where p_y is the ratio of examples in D with $y=1$



- **At test time**: $x \longrightarrow$ leaf t, predict majority class of D_t
- **Goal**: make the distribution of y in each leaf highly unbalanced $\longrightarrow$ helps classification

FARE: Decision Tree Modifications

We modify the procedure in two ways to make the tree **fairness-aware**:

(1) Fairness-aware criterion – instead of $Gini_y(D)$ use the following:

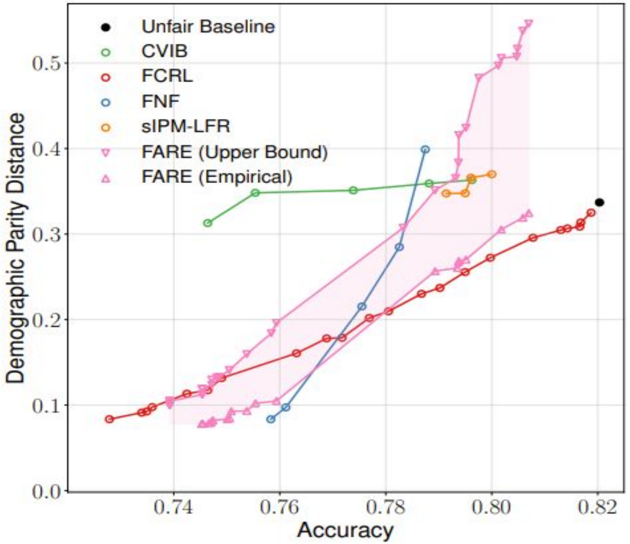
$$FairGini(D) = (1-\gamma)Gini_y(D) + \gamma(0.5 - Gini_s(D)) \in [0, 0.5]$$

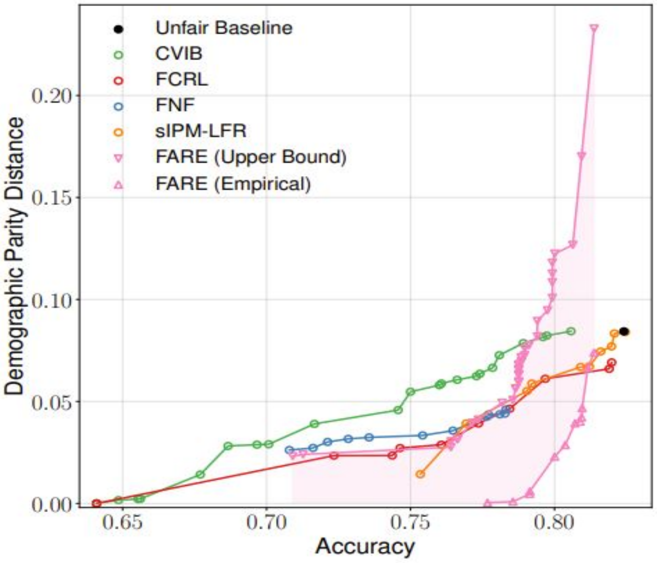
- Makes the distribution of y in each leaf **highly unbalanced** (as before)
- Makes the distribution of s in each leaf **uniform** (to prevent the adversary from distinguishing s); parameter γ controls the accuracy/fairness tradeoff

(2) Fairness-aware categorical splits – generalization of the Breiman shortcut for efficient heuristic treatment of categorical variables (*see the paper*)

FARE: Results

- Similar **empirical** accuracy/fairness tradeoff as prior methods
- **Provable unfairness upper bound** with no restrictive assumptions





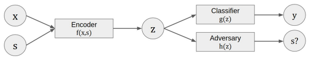
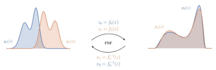
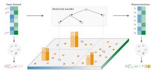
Future Work

- Other classes of restricted encoders?
- Investigate the new 3-way **accuracy/fairness/bound tightness** tradeoff?
- Can we adapt this to **other domains** (e.g., images, text, graphs)? (LCIFR $\longrightarrow$ LASSI)
- **FRL Benchmark**: literature is out of sync in terms of evaluation procedures
  1) Common usage of old, small-scale datasets with known issues
  2) No agreement on a set of fairness constraints or the procedure for training downstream classifiers (which can greatly affect results)
  3) No single source of truth for state-of-the-art methods (common in other fields, e.g., RobustBench for adversarial robustness)

Lecture Summary

Ensuring group fairness: **pre-processing (FRL)** vs **in-training** vs **post-processing** methods

Three methods for group-fair representation learning:

	Method Class	**Encoder Type**	**Assumptions**
LAFTR 	Theoretically-principled FRL (no guarantees)	Neural Networks	/
FNF 	**Provable** FRL	Normalizing Flows	Density Estimation with Guarantees
FARE 	**Provable** FRL	Restricted Encoders	/