Reliable and Trustworthy Artificial Intelligence

Lecture 11: Individual Fairness

Anian Ruoss

DeepMind

Fall 2022

http://www.sri.inf.ethz.ch

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Individual Fairness – Connecting the Dots

We will learn how to train models that *provably satisfy individual fairness* by combining many techniques we have seen so far:

- Robustness certification
    - MILP
    - Convex relaxations
    - Randomized smoothing
- Training neural networks with logic

Moreover, we will also leverage other recent advances in machine learning:

- Representation learning
- Generative modeling

Recap: What Does It Mean to Be Fair?

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: Individual Fairness

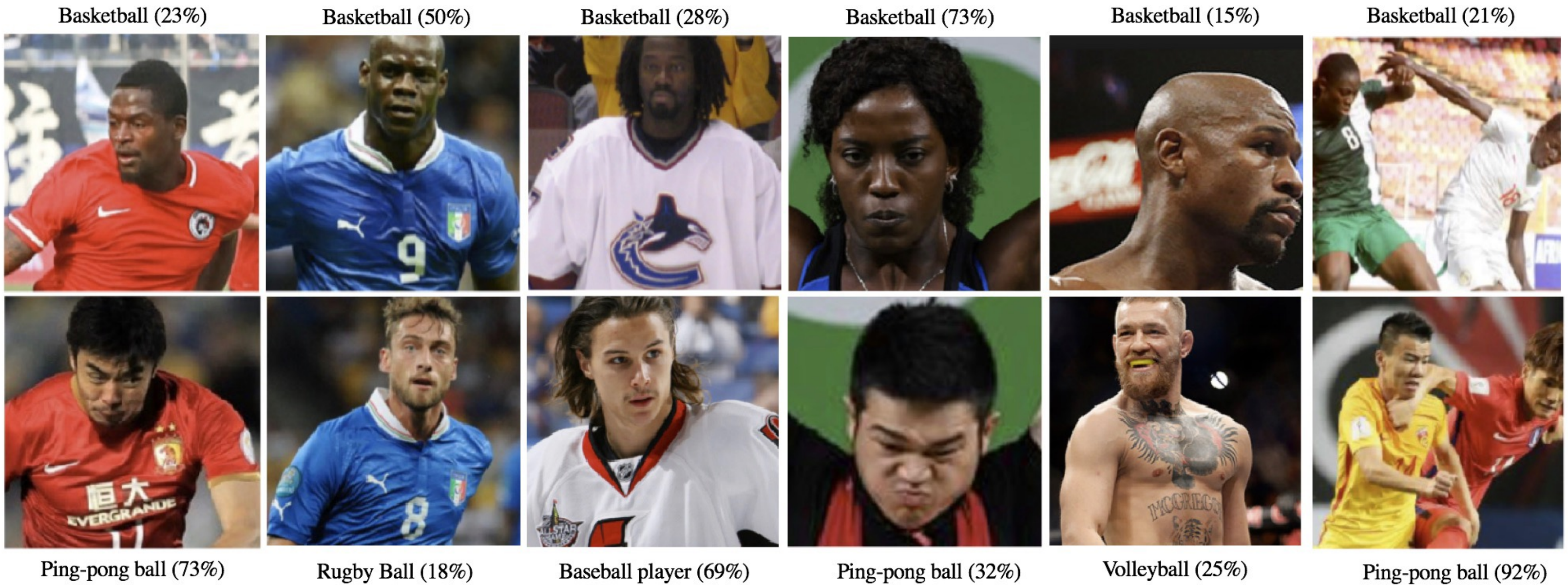
Definition (Individual Fairness): A model $M: R^n \rightarrow R^o$ is *individually fair* if for two data points x and x' that are similar to each other (according to some input similarity notion ϕ) it produces similar outputs $M(x)$ and $M(x')$ (according to some output similarity notion μ).

Examples

- Lipschitz mapping: A mapping $M : R^n \rightarrow R^o$ satisfies the (μ, ϕ)-Lipschitz property if for every $x, x' \in X$ we have $\mu\big(M(x), M(x')\big) \leq \phi(x, x')$.

- Binary similarity metrics: For binary input and output similarity metrics, i.e., $\phi: R^n \times R^n \rightarrow \{0, 1\}$ and $\mu: R^o \times R^o \rightarrow \{0, 1\}$, we can reformulate individual fairness for the points x and x'as $\phi(x, x') \Rightarrow \mu\big(M(x), M(x')\big)$.

Why Individual Fairness?



Machine learning models have been shown to classify pairs of similar individuals playing the same sport differently (Stock and Cisse, ECCV'18).

Why Individual Fairness?

Unfortunately, a model that satisfies *group fairness does not necessarily satisfy individual fairness*.

Example

ID	a (e.g., gender)	x (e.g., income)	$M([x, a])$ (e.g., loan decision)
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

M satisfies demographic parity: $P(M(\cdot) = 1|a = 0) = P(M(\cdot) = 1\ |a = 1) = \frac{1}{2}$.

However, for the similarity metric $\phi([x, a], [x', a']) = 1 \iff x = x'$, the individuals {0,2} and {1,3} are treated differently even though they are similar.

Problems with Individual Fairness

How can we *define a suitable similarity metric* (requires significant domain expertise and human insight)?

- Define a metric, e.g., norm, in the feature space (Dwork et al., 2012).
- Learn a metric from data (Mukherjee et al., ICML'20).
- Learn from human feedback (Dorner et al., NLP'23).

How can we *train/modify* a model such that it *satisfies individual fairness without compromising downstream accuracy*?

- Pre-processing: Debias the data, such that standard training yields a fair model.
- In-processing: Change the training pipeline to learn a fair model on biased data.
- Post-processing: Modify model predictions during inference time.

How can we *guarantee* that the model is actually individually fair?

Individual Fairness as a Robustness Problem

Consider the Lipschitz formulation of individual fairness with the Lipschitz constant $L > 0$, given by

$$\mu\big(M(x), M(x')\big) \leq L\ \phi(x, x').$$

Moreover, consider $\phi(x, x') \coloneqq (x - x')^T S(x - x')$, with a symmetric positive definite covariance matrix S (Mukherjee et al., ICML'19).

Finally, let M be a classifier and $\mu\big(M(x), M(x')\big) \coloneqq [M(x) \neq M(x')]$.

Then, using the Mahalanobis norm $||x||_S \coloneqq \sqrt{x^T S x}$, we need to show that

$$[M(x) \neq M(x')] \leq L||x\ - x'||_S^2 \quad \forall x'$$

$$\Leftrightarrow$$

$$||x\ - x'||_S^2 < \frac{1}{L} \Rightarrow M(x) = M(x') \quad \forall x'$$

$$\Leftrightarrow$$

$$M(x) = M(x + \delta) \quad \forall \delta\ with\ ||\delta||_S^2 < \frac{1}{L}$$

$$\Leftrightarrow$$

$$M(x) = M(x + \delta) \quad \forall \delta\ with\ ||\delta||_S < \frac{1}{\sqrt{L}}$$

Individual Fairness as a Robustness Problem

Finally, we observe that $M(x) = M(x + \delta), \forall \delta\ with\ ||\delta||_S \leq \frac{1}{\sqrt{L}}$ is the same as saying that M needs to be robust to adversarial perturbations of magnitudes up to $\frac{1}{\sqrt{L}}$ with respect to the $|| \cdot ||_S$ norm.

Key Insight: We can reformulate individual fairness as a local robustness problem, allowing the transfer of many established techniques.

As a result, we just have to choose a suitable class of similarity metrics that allow us to perform standard robustness certification, e.g., norms (Yeom & Fredrikson, IJCAI'20) or logical constraints (Ruoss et al., NeurIPS'20).

A Real-World Scenario

Consider a manager overseeing different teams, all using the same data to build predictive models for different products (Cisse & Koyejo, 2019). The manager seeks to *ensure both fairness and accuracy* across the products.

However, each team is solving a different prediction task. Moreover, there is no company policy on fairness, thus no shared guidelines.

- Team *alpha* is fully focused on accuracy, but is oblivious about fairness issues.
- Teams *beta*, *nu*, and *gamma* are all interested in fairness. Each team is excited to implement this but each team has selected different fairness definitions.
- Team *zeta* would like to improve the fairness of their predictions, but has no idea how to incorporate or measure fairness.
- The *manager* wants to independently verify that all released products are fair.

A Real-World Scenario

Consider a manager overseeing different teams, all using the same data to build predictive models for different products (Cisse & Koyejo, 2019). The manager seeks to *ensure both fairness and accuracy* across the products.

Challenges

- Some teams do not have the expertise (or interest) to design fairer models.
- Different teams use different definitions of fairness.
- Incorporating fairness can have different impacts on the performance of the models across the products.
- Auditing all the predictive models for fairness can be challenging when each team has its own recipe.

Fair Representation Learning to the Rescue

Fair representation learning (Zemel et al., ICML’13; McNamara et al., AIES’19) partitions the process of training fair models into three parties:

Data Regulator

Determines fairness criteria and data source(s), audits results.

Data Producer

Computes the fair representation given data regulator criteria.

Data Consumer

Trains the ML model given the sanitized data.

Fair Representation Learning to the Rescue

In the fair representation learning setting, the model $M: R^n \rightarrow R^o$ is composed of two parts:

- an encoder $f_\theta: R^n \rightarrow R^k$, provided by the data producer
- a predictor $h_\psi: R^k \rightarrow R^o$, provided by the data consumer

with R^k denoting the latent space. The model is thus defined as $M = h_\psi \circ f_\theta$.

This induces modularity: The data consumer can train its model without worrying about fairness if the representation provided by the producer is fair.

⇒ The data consumer *can be ignorant of the fairness concerns* in the system!

Fair Representation Learning

Pros

- Often more efficient than alternatives (especially with re-use)
- Can be employed when the data user is untrusted or apathetic about fairness
- Inherits good properties from representation learning (e.g., transfer learning)
- Audits can be more efficient (especially when only auditing the representation)

Cons

- Less precise control of the fairness/performance tradeoff (than joint training)
- May lead to fairness overconfidence (data consumer acts adversarially)
- Startup costs can be high (representation learning can be expensive)

Bernie
I am once again asking
for provable certificates

Fair Representation Learning with Guarantees

Can we augment the fair representation learning framework in a minimally invasive way to obtain provable certificates of individual fairness?

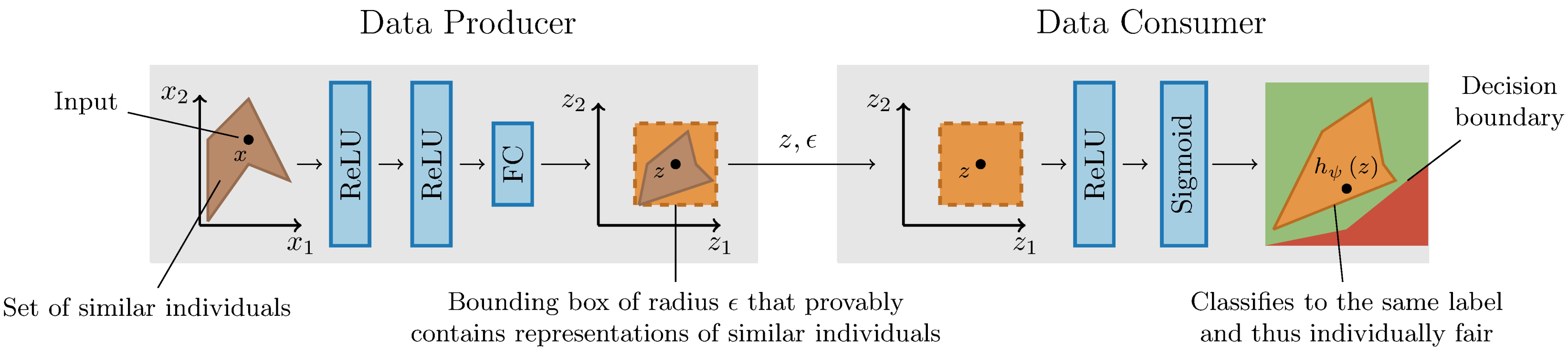
We want to *maintain the modularity*, i.e., the data consumer should not have to care about fairness and should be able to use standard training techniques that achieve good performance on downstream tasks.

Moreover, the data regulator should be able to *formally verify that individual fairness holds* across the entire model pipeline.

LCIFR to the Rescue

Learning certified individually fair representations (Ruoss et al., 2020) is a framework that:

- allows the data consumer to be oblivious of fairness
- allows the data consumer to use standard training techniques
- allows the data producer to be oblivious of the downstream task
- allows the data regulator to define similarity via interpretable logical constraints
- allows the data regulator to certify the fairness of the end-to-end model



Data Producer
Data Consumer
Input
x_2
x
x_1
ReLU
ReLU
FC
z_2
z
z_1
z, ϵ
z_2
z
z_1
ReLU
Sigmoid
$h_\psi(z)$
Decision boundary
Set of similar individuals
Bounding box of radius ϵ that provably contains representations of similar individuals
Classifies to the same label and thus individually fair

LCIFR: The Data Regulator

We consider binary input and output similarity measures $\phi: R^n \times R^n \rightarrow \{0, 1\}$ and $\mu: R^o \times R^o \rightarrow \{0, 1\}$, that can be expressed in a rich logical fragment, i.e., anything that DL2 (Fischer et al., ICML'19) and MILP (Tjeng et al., ICLR'19) can handle.

By working with declarative constraints, *data regulators can express interpretable, domain-specific notions of similarity*.

Example

$$\phi(x, x') \coloneqq \bigwedge_{i \in Categorical \setminus \{race, gender\}} (x_i = x_i') \bigwedge_{j \in Numerical} |x_j - x_j'| \leq \alpha$$

Importantly, logical constraints also capture categorical features exactly (unlike, e.g., ℓ_p-norms), which are prevalent in fairness datasets.

LCIFR: The Data Regulator

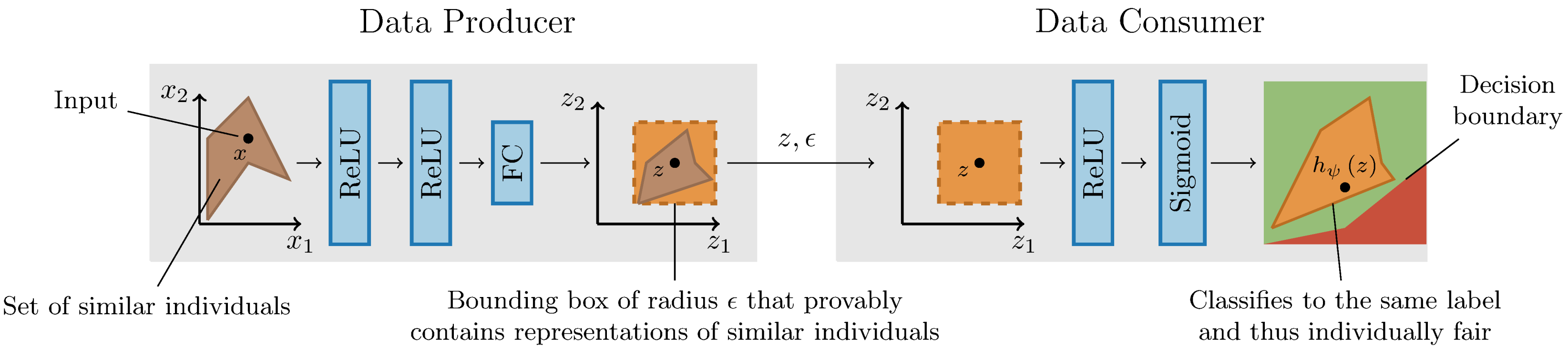
The binary input similarity metric induces a set $S_\phi(x) \coloneqq \{x' \in R^N \mid \phi(x, x')\}$ of all individuals similar to x. The data regulator wants to *certify* for individual x in the test dataset,

$$\forall x' \in S_\phi(x) \Longrightarrow \mu\big(M(x), M(x')\big).$$

If we consider classification with $\mu\big(M(x), M(x')\big) \Leftrightarrow M(x) = M(x')$, we thus want to maximize the number of data points x from the test set for which we can certify that the set of all similar individuals $S_\phi(x)$ is classified the same.

How can we achieve this in the fair representation learning setting?

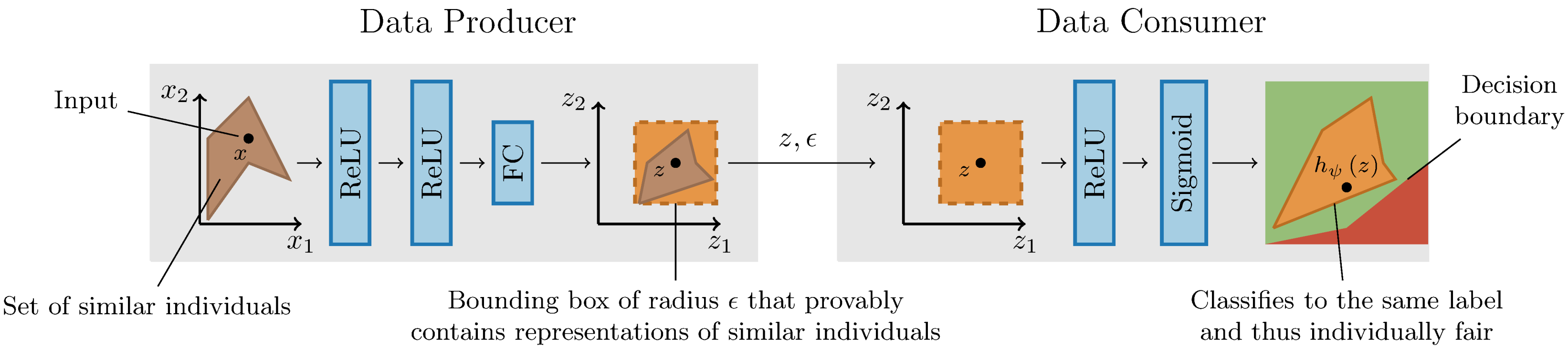
LCIFR: The Data Producer


Data Producer
Data Consumer
Input
x_2
x
x_1
ReLU
ReLU
FC
z_2
z
z_1
z, ϵ
z_2
z
z_1
ReLU
Sigmoid
$h_\psi(z)$
Decision boundary
Set of similar individuals
Bounding box of radius ϵ that provably contains representations of similar individuals
Classifies to the same label and thus individually fair

The data producer:

- trains the encoder $f_\theta : R^n \to R^k$ to map all similar points close together in latent space, i.e., $\forall x' \in S_\phi(x) \implies |f_\theta(x) - f_\theta(x')|_\infty \leq \delta$ using DL2's translation of logical constraints to a differentiable loss function
- encodes $S_\phi(x)$ and f_θ as mixed-integer linear programs (MILP) to compute ϵ such that $f_\theta\left(S_\phi(X)\right) \subseteq \{z' | |z - z'|_\infty \leq \epsilon\}$ for $z := f_\theta(x)$

LCIFR: The Data Consumer


Data Producer
Data Consumer
Input
x_2
x
x_1
ReLU
ReLU
FC
z_2
z
z_1
z, ϵ
z_2
z
z_1
ReLU
Sigmoid
$h_\psi(z)$
Decision boundary
Set of similar individuals
Bounding box of radius ϵ that provably contains representations of similar individuals
Classifies to the same label and thus individually fair

The data consumer:

- obtains z and ϵ from the data producer (but not the original data x) and uses local robustness training to make the classifier $h_\psi : R^k \rightarrow R^o$ robust against l_∞-perturbations of magnitude ϵ around z
- uses neural network robustness verifier to certify ϵ-robustness for z

LCIFR: The Optimization Problems

Data Producer

We want to translate the encoder constraint $\phi(x, x') \Longrightarrow |f_\theta(x) - f_\theta(x')|_\infty \ \leq \ \delta$ into a differentiable loss function $L(\phi)$ such that $L(\phi)(x, x') = 0$ if and only if the implication is satisfied (using DL2).

Denoting $\omega(x, x') \coloneqq |f_\theta(x) - f_\theta(x')|_\infty \ \leq \ \delta$, we have

$$L(\phi \Rightarrow \omega) = L(\neg\phi \vee \omega) = L(\neg\phi) \cdot L(\omega).$$

Moreover, we have

$$L(\omega)(x, x') = L(|f_\theta(x) - f_\theta(x')|_\infty \ \leq \ \delta) = \max\{|f_\theta(x) - f_\theta(x')|_\infty \ - \ \delta, 0\}.$$

LCIFR: The Optimization Problems

Data Producer

Using this differentiable loss, the data producer can now approximate the problem of finding an encoder f_θ that maximizes the probability that $\phi \Rightarrow \omega$ via the following min-max optimization problem:

First, we find a counterexample

$$x^* = \underset{x' \in S_\phi(x)}{\arg\min}\, L\,(\neg(\phi \Rightarrow \omega))(x, x')$$

Then, we find the parameters θ that minimize the constraint loss at x^*

$$\underset{\theta}{\arg\min}\, E_{x\sim D}[L\,(\phi \Rightarrow \omega)(x, x^*)]$$

LCIFR: The Optimization Problems

Data Producer

To ensure the modularity between the data producer and consumer, the latent representation needs to remain informative for downstream applications.

To that end, the data producer additionally trains a classifier $\mathrm{q}: \mathrm{R}^{\mathrm{k}} \rightarrow R^{o}$ that tries to predict the target label y from the latent representation $z = f_\theta(x)$.

Thus, the data producer jointly trains the encoder f_θ and classifier q to minimize the combined objective

$$\underset{f_\theta, q}{\arg\min}\, E_{x,y}[L_C\big(q\big(f_\theta(x)\big), y\big) + \gamma L_F(x, f_\theta(x))]$$

where $\mathrm{L_C}$ is a classification loss (e.g., cross-entropy or a task-agnostic transfer loss), L_F is the fairness loss obtained via DL2, and γ balances the two objectives.

LCIFR: The Optimization Problems

Data Consumer

Assuming that the encoder f_θ has been trained to maintain predictive utility and satisfy the fairness constraint, the data consumer only needs to ensure local robustness of the classifier h_ψ to achieve individual fairness.

This can be achieved via standard min-max optimization

$$\arg\min_{\psi} E_{z\sim D_z}[\max_{\pi\in[\pm\delta]} L_C(h_\psi(z+\pi), y)]$$

where D_z is the latent distribution obtained by sampling from D and applying the encoder f_θ.

LCIFR: The Open Problems

We can learn certified individually fair representations for low-dimensional data.

However, real-world models operate on *high-dimensional inputs* (e.g., images). For such data, input similarity cannot be measured in terms of the features (e.g., by comparing pixels), meaning that we cannot apply our logical constraints.

Moreover, standard neural network verifiers *do not scale* to real-world models.

Challenges

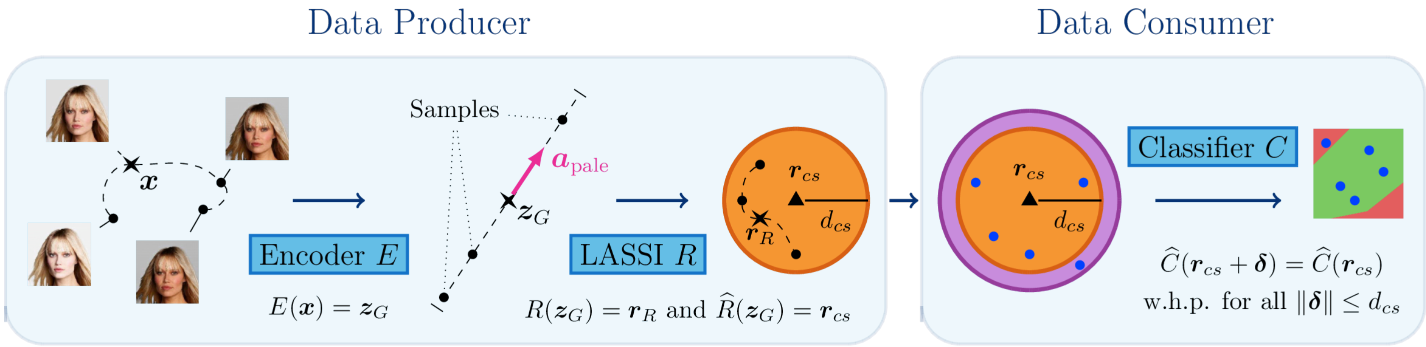
- designing a suitable input similarity metric for high-dimensional data
- scaling fairness certification to real-world models

LASSI to the Rescue

Latent space smoothing for individually fair representations (Peychev et al., ECCV’22) is a framework that extends LCIFR to the high-dimensional setting.

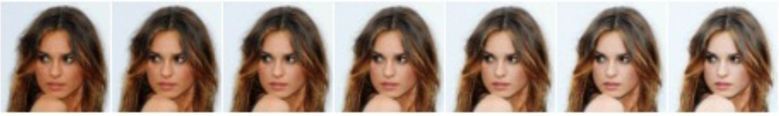
Concretely, it allows the data regulator to formulate constraints such as

For a given person, all people differing only in their hair color should receive the same classification output.

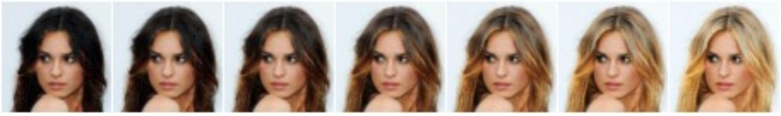

Data Producer
Data Consumer
$\boldsymbol{x}$
Encoder E
$E(\boldsymbol{x}) = \boldsymbol{z}_G$
Samples
$\boldsymbol{a}_{\text{pale}}$
$\boldsymbol{z}_G$
LASSI R
$R(\boldsymbol{z}_G) = \boldsymbol{r}_R$ and $\widehat{R}(\boldsymbol{z}_G) = \boldsymbol{r}_{cs}$
$\boldsymbol{r}_{cs}$
d_{cs}
$\boldsymbol{r}_R$
Classifier C
$\widehat{C}(\boldsymbol{r}_{cs} + \boldsymbol{\delta}) = \widehat{C}(\boldsymbol{r}_{cs})$
w.h.p. for all $\|\boldsymbol{\delta}\| \leq d_{cs}$

Generative Modeling

We leverage *generative models* to define the set of similar individuals in the latent space of the model by varying a continuous attribute of the image.



Varying the *“pale skin”* attribute.



Varying the *“blond hair”* attribute.

We consider generative models consisting of an encoder $E\colon \mathrm{R}^{\mathrm{n}} \rightarrow R^{q}$ and a decoder $D\colon \mathrm{R}^{\mathrm{q}} \rightarrow R^{n}$ with $z_G = E(x)$.

Then, the data regulator defines the set of similar individuals for point x in the latent space of the generative model using the attribute vector a_{pale}:

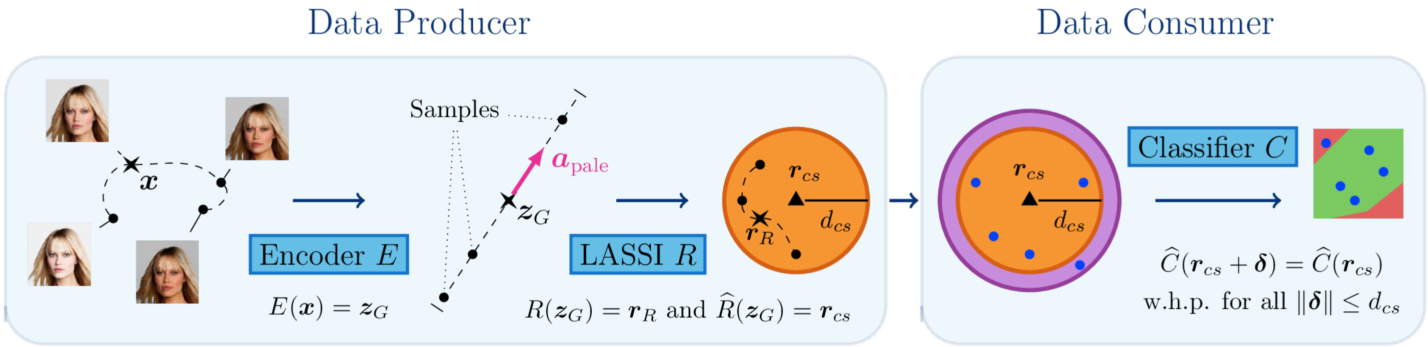
$$S(x) \coloneqq \left\{ z_G + t \cdot a_{pale} \,\middle|\, t \in [-\epsilon, \epsilon] \right\}$$

Generative Modeling

How can we compute meaningful attribute vectors, such as a_{pale}?

For example, compute the average latent vectors $z_{G,pale}$ and $z_{G,\neg pale}$ and use their difference, i.e., $a_{pale} \coloneqq z_{G,pale} - z_{G,\neg pale}$ (Kingma & Dhariwal, 2018).

Note that LASSI is independent of the actual computation of the attribute vector.


Data Producer
Data Consumer
$\boldsymbol{x}$
Encoder E
$E(\boldsymbol{x}) = \boldsymbol{z}_G$
Samples
$\boldsymbol{a}_{\text{pale}}$
$\boldsymbol{z}_G$
LASSI R
$R(\boldsymbol{z}_G) = \boldsymbol{r}_R$ and $\widehat{R}(\boldsymbol{z}_G) = \boldsymbol{r}_{cs}$
$\boldsymbol{r}_{cs}$
d_{cs}
$\boldsymbol{r}_R$
Classifier C
$\widehat{C}(\boldsymbol{r}_{cs} + \boldsymbol{\delta}) = \widehat{C}(\boldsymbol{r}_{cs})$
w.h.p. for all $\|\boldsymbol{\delta}\| \leq d_{cs}$

Recap: Randomized Smoothing

Unlike MILP, randomized smoothing (Cohen et al., ICML'19) can compute local robustness guarantees for any type of classifier $C: R^k \to Y$, regardless of its complexity and scale.

To that end, we construct a smoothed classifier $\hat{C}: R^k \to Y$, which returns the most probable classification for and input $r \in R^k$ when perturbed by random noise from $N(0, \sigma_{rs}^2 I)$.

Moreover, using a sampling-based approach, we can establish a local robustness guarantee of the form $\forall \delta \in R^k$ such that $||\delta||_2 < d_{rs}$ we have $\hat{C}(r + \delta) = \hat{C}(r)$ with probability $1 \; - \alpha_{rs}$, where α_{rs} can be made arbitrarily small.

TL;DR

Randomized smoothing yields $\hat{C}$ that will classify all points in the ℓ_2-ball radius d_{rs} around r the same with high probability.

Center Smoothing

Center smoothing (Kumar & Goldstein, NeurIPS’21) extends randomized smoothing to the multidimensional regression setting.

Concretely, for a function $R: R^q \rightarrow R^k$, center smoothing uses sampling and approximation to compute a smooth version $\hat{R}: R^q \rightarrow R^k$ that maps $z \in R^q$ to the center point $r_{cs} \coloneqq \hat{R}(z)$ of a minimum enclosing ball containing at least half of the points $r_i \sim R(z + N(0, \sigma_{cs}^2 I))$ for $i \in \{1, \dots, m\}$.

Then, for $\epsilon > 0$ and $\forall z' \in R^q$ such that $||z - z'||_2 \leq \epsilon$, we have $||\hat{R}(z) - \hat{R}(z')||_2 \leq d_{cs}$ with probability at least $1 - \alpha_{cs}$.

TL;DR

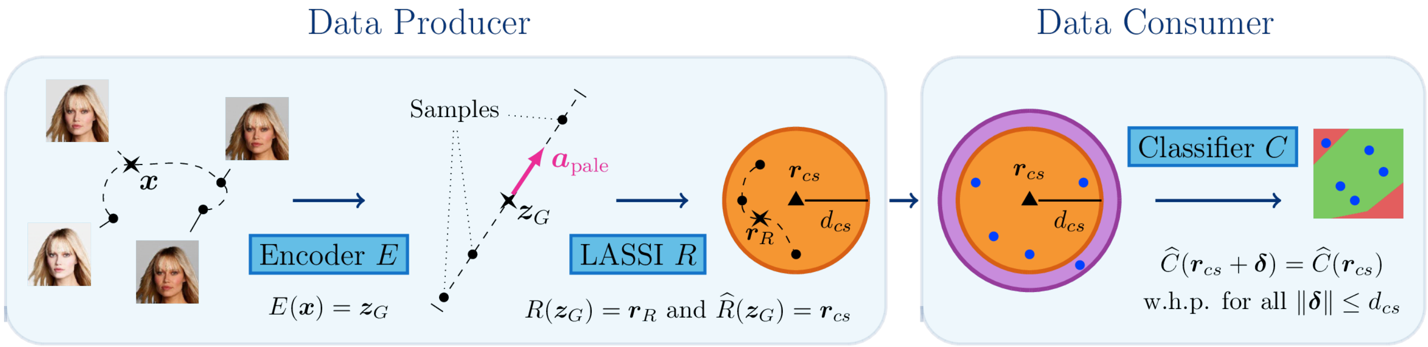
Center smoothing computes a sound upper bound d_{cs} on the ℓ_2-ball of the function outputs of $\hat{R}$ for all points in the ℓ_2-ball of radius ϵ around z.

LASSI: Putting the Pieces Together

The data producer uses center smoothing to compute a representation that provably maps all similar points close together with high probability.

The data consumer uses randomized smoothing to certify that all points within a certain radius get classified the same with high probability.

Theorem: The end-to-end certificate of individual fairness holds with probability $1 - \alpha_{rs} - \alpha_{cs}$ (union bound).

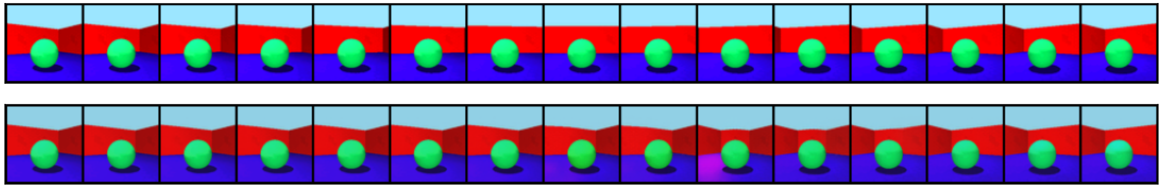

Data Producer
Data Consumer
$\boldsymbol{x}$
Encoder E
$E(\boldsymbol{x}) = \boldsymbol{z}_G$
Samples
$\boldsymbol{a}_{\text{pale}}$
$\boldsymbol{z}_G$
LASSI R
$R(\boldsymbol{z}_G) = \boldsymbol{r}_R$ and $\widehat{R}(\boldsymbol{z}_G) = \boldsymbol{r}_{cs}$
$\boldsymbol{r}_{cs}$
d_{cs}
$\boldsymbol{r}_R$
$\boldsymbol{r}_{cs}$
d_{cs}
Classifier C
$\widehat{C}(\boldsymbol{r}_{cs} + \boldsymbol{\delta}) = \widehat{C}(\boldsymbol{r}_{cs})$
w.h.p. for all $\|\boldsymbol{\delta}\| \leq d_{cs}$

Problems with LASSI

The validity of our fairness certificate depends heavily on the generative model.

Concretely, the similarity sets $S(x)$ may not be exhaustive enough as there can be latent points outside $S(x)$ that correspond to input points that would be perceived as similar to x by a human observer. Thus, we may certify a model to be fair without considering all relevant similar individuals.

In general, it is hard to obtain formal guarantees for the generative model that transfer to the real world (there is no ground truth data).



Initial experiments on procedurally generated datasets indicate that the certificates may transfer to the real world.

Lecture Summary

- We learned how to enforce fairness in a modular manner via fair representation learning.

- We learned how to augment fair representation learning with provable certificates of individual fairness by combining DL2 and MILP.

- We learned how to augment fair representation learning with provable certificates of individual fairness for high-dimensional data by combining generative modeling and randomized smoothing.