How Biased is Your Algorithm? Auditing and Explaining Unfairness in ML

Debabrota Basu[a]

Workshop@Comète on Ethical AI, Inria Saclay

[a]Équipe Scool, Univ. Lille, Inria, UMR 9189-CRIStAL, CNRS, Centrale Lille, France

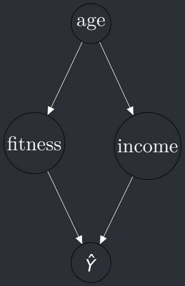
Outline

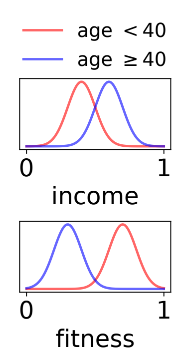
1. Motivation: Auditing, Understanding, and Eliminating Bias
2. Fairness Verification: Boolean Formulas with Independent Features
3. Fairness Explanation: A Model-agnostic Approach
4. Curtain Call: Question Avenir

(Un)Fairness in Machine Learning

Prediction of eligibility of health insurance

- Sensitive features, $\mathbf{A} = \{\text{age}\}$
- Non-sensitive features, $\mathbf{X} = \{\text{fitness, income}\}$

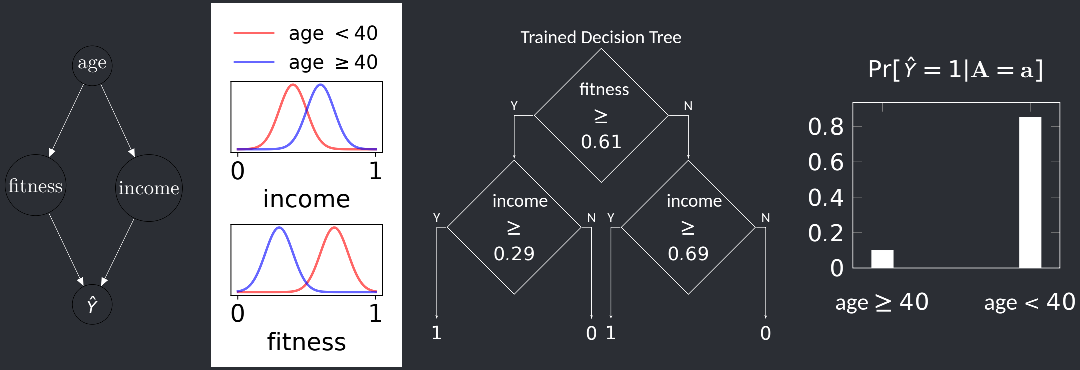




(Un)Fairness in Machine Learning

Prediction of eligibility of health insurance

- Sensitive features, $\mathbf{A} = \{\text{age}\}$
- Non-sensitive features, $\mathbf{X} = \{\text{fitness, income}\}$



Final Destination

Auditing, Understanding, and Eliminating Bias

- **Problem:** ML classifiers may become unfair/biased to certain demographic groups
- **Solution:** Multiple fairness metrics & algorithms are proposed to enhance fairness
- **Missing Link:** Scalable algorithms for verification and explanation of fairness

Plat du Jour

- **Fairness Verification**: A rigorous estimate of fairness of a classifier
- **Fairness Explanation**: Identifying the source of unfairness of a classifier through the lens of input features

Outline

1. Motivation: Auditing, Understanding, and Eliminating Bias
2. Fairness Verification: Boolean Formulas with Independent Features
3. Fairness Explanation: A Model-agnostic Approach
4. Curtain Call: Question Avenir

Justicia: A Stochastic SAT Approach to Formally Verify Fairness

Fairness Verification with Boolean Representation [GBM21]

Given

- a binary classifier $\mathscr{A} : (\mathbf{X}, \mathbf{A}) \rightarrow \hat{Y} \in \{0, 1\}$ and
- a probability distribution $(\mathbf{X}, \mathbf{A}, Y) \sim \mathscr{D}$,

verify whether $\mathscr{A}$ achieves fairness w.r.t. $\mathscr{D}$

Statistical parity: $\mathscr{A}$ satisfies ϵ-statistical parity if for $\epsilon \in [0, 1]$,

$$\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] - \min_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] \leq \epsilon$$

Justicia: A Stochastic SAT Approach to Formally Verify Fairness

Fairness Verification with Boolean Representation [GBM21]

Given

- a binary classifier $\mathscr{A} : (\mathbf{X}, \mathbf{A}) \rightarrow \hat{Y} \in \{0, 1\}$ and
- a probability distribution $(\mathbf{X}, \mathbf{A}, Y) \sim \mathscr{D}$,

verify whether $\mathscr{A}$ achieves fairness w.r.t. $\mathscr{D}$

Key Quantity

$\Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$ is called the conditional PPV (Positive Predictive Value)

Statistical parity: $\mathscr{A}$ satisfies ϵ-statistical parity if for $\epsilon \in [0, 1]$,

$$\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] - \min_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] \leq \epsilon$$

Our Approach: Compute the maximum and minimum of $\Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$ by *a reduction to stochastic SAT*

Satisfiability (SAT) problem

A Recap

Given a Boolean formula ϕ in CNF (Conjunctive Normal Form) defined over Boolean variables $\mathbf{X}$, the SAT problem finds a satisfying assignment of $\mathbf{X}$ that evaluates ϕ to true

$$\phi = (X_1 \lor \neg X_2) \land (\neg X_1 \lor X_2 \lor X_3) \land \neg X_1$$

- SAT solution: $X_1 = \text{false}$, $X_2 = \text{false}$, $X_3 = \text{true}$

Stochastic SAT (SSAT)

A Brief Introduction

An SSAT formula Φ has a prefix and a CNF formula ϕ

$$\Phi = \underbrace{q_1 X_1, \ldots, q_n X_n}_{\text{prefix}}, \phi$$

- q_i is an universal ($\forall$), existential ($\exists$), or randomized $\reflectbox{R}^{p_i}$ quantifier with $p_i = \Pr[X_i = \text{ true}]$
- SSAT computes the probability of satisfaction $\Pr[\Phi]$

Stochastic SAT (SSAT)

The Semantics

Let X be the left-most variable in the prefix of Φ. The recursive semantics of a SSAT formula are

1. $\Pr[\text{true}] = 1, \Pr[\text{false}] = 0$
2. $\Pr[\Phi] = \max_X\{\Pr[\Phi|_X], \Pr[\Phi|_{\neg X}]\}$ if X is existentially quantified ($\exists$)
3. $\Pr[\Phi] = \min_X\{\Pr[\Phi|_X], \Pr[\Phi|_{\neg X}]\}$ if X is universally quantified ($\forall$)
4. $\Pr[\Phi] = p\Pr[\Phi|_X] + (1-p)\Pr[\Phi|_{\neg X}]$ if X is randomized quantified ($\reflectbox{R}^p$)

Stochastic SAT (SSAT)

A Tale of Two Encodings

- **Existential-random SSAT formula**

$$\Phi_{\mathrm{ER}} = \exists X_2, \exists X_3, \rotatebox[origin=c]{180}{R}^{0.25} X_1,\ (X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge \neg X_1$$

  - $\Pr[\Phi_{\mathrm{ER}}] = 0.75$
  - Optimal assignment (maximization): $X_2 = \text{false}$, $X_3 = \text{false}$

- **Universal-random SSAT formula**

$$\Phi_{\mathrm{UR}} = \forall X_2, \forall X_3, \rotatebox[origin=c]{180}{R}^{0.25} X_1,\ (X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge \neg X_1$$

  - $\Pr[\Phi_{\mathrm{UR}}] = 0$
  - Optimal assignment (minimization): $X_2 = \text{true}$, $X_3 = \text{false}$

Justicia: Fairness Verification with SSAT

- features $\mathbf{X} \cup \mathbf{A}$ are Boolean
- predicted class $\hat{Y}$ is a CNF formula $\phi_{\hat{Y}}$ defined on $\mathbf{X} \cup \mathbf{A}$

Two Steps to Justicia

1. Computing $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, is equivalent to solving

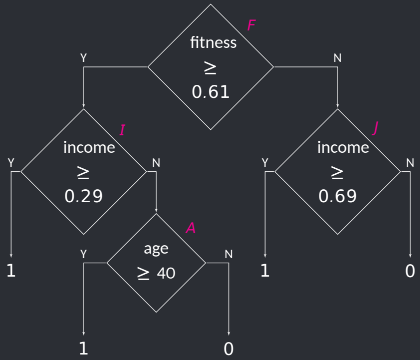
$$\Phi_{\mathrm{ER}} \triangleq \underbrace{\exists A_1, \ldots, \exists A_n,}_{\text{sensitive features}} \underbrace{\text{Я}^{p_1} X_1, \ldots, \text{Я}^{p_m} X_m,}_{\text{non-sensitive features}} \phi_{\hat{Y}}.$$

2. For computing $\min_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, we substitute $\exists$ with $\forall$ for sensitive features, and observe $\Pr[\Phi_{\mathrm{UR}}] = 1 - \Pr[\Phi_{\mathrm{ER}}(\neg \phi_{\hat{Y}})]$.

Use an SSAT solver to solve the ER-SSAT problems [LWJ18].

An Illustration

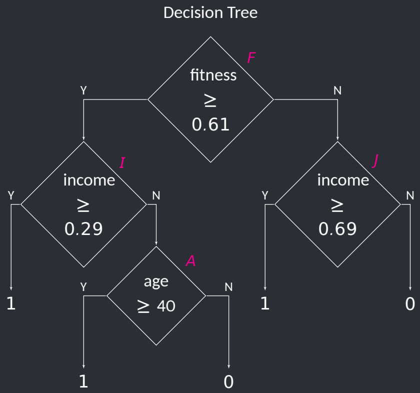
Decision Tree



- CNF representation: $(\neg F \vee I \vee A) \wedge (F \vee J)$
- $\Pr[F] = 0.41, \Pr[I] = 0.93, \Pr[J] = 0.09$
- To compute $\max_{\mathrm{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, we construct

$$\Phi_{ER} = \exists A, \text{Я}^{0.41} F, \text{Я}^{0.93} I, \text{Я}^{0.09} J,\ (\neg F \vee I \vee A) \wedge (F \vee J)$$

An Illustration



- CNF representation: $(\neg F \vee I \vee A) \wedge (F \vee J)$
- $\Pr[F] = 0.41, \Pr[I] = 0.93, \Pr[J] = 0.09$
- To compute $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, we construct

$$\Phi_{ER} = \exists A, \text{Я}^{0.41} F, \text{Я}^{0.93} I, \text{Я}^{0.09} J, \ (\neg F \vee I \vee A) \wedge (F \vee J)$$

- $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] = \Pr[\Phi_{ER}] = 0.46$
- $\min_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] = 0.43$
- Statistical parity is $0.46 - 0.43 = 0.03$

Theoretical Analysis

Psuedologarithmic Sample Complexity

Theorem (A PAC Bound for Justicia)

With probability $1-\delta$, Justicia can estimate Statistical Parity (SP) up to a multiplicative error $2\epsilon_0$, i.e. $\widehat{\mathrm{SP}} \leq 2\epsilon_0 \mathrm{SP}$, if it has access to

$$k = O\left(\left(n + \ln\left(\frac{1}{\delta}\right)\right)\frac{\ln m}{\ln \epsilon_0}\right)$$

samples from the data-generating distribution.

Here, m and n are the number of variables with randomised and existential quantifiers respectively. Note that $\delta \in (0, 1)$ and $\epsilon_0 > 1$.

Experimental Analysis

Robustness and Compound Attribute Level Analysis

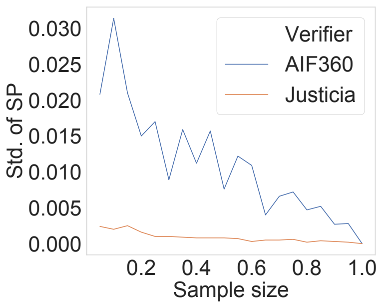


Figure: Robustness between probabilistic (Justicia) and dataset centric (AIF360 [BDH+18]) verifiers

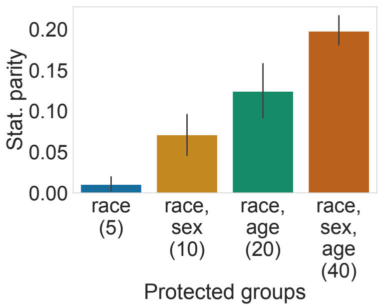


Figure: Verifying compound sensitive/protected groups with Justicia

Experimental Results

Faster than the Fastest

State-of-the-art probabilistic fairness verifiers

- FairSquare: computes weighted volume of programs using SMT reduction [ADDN17]
- VeriFair: probabilistic verification via sampling [BZSL19]

Dataset	FairSquare	VeriFair	Justicia
Ricci	4.8	5.3	0.1
Titanic	16	1.2	0.1
COMPAS	36.9	15.9	0.1
Adult	—	295.6	0.2

Table: Runtime of different verifiers in terms of execution time (in seconds) with decision tree classifiers. '—' refers to timeout.

Summary of Justicia [GBM21]

What Justicia can do?

- Justicia is a SSAT based probabilistic fairness verifier
- First method to verify compound sensitive groups
- More scalable in verifying decision trees and classifiers in Boolean formulas

What Justicia cannot do?

- Classifiers have to be expressed as Boolean formulas, which is computationally expensive even for linear classifiers
- Assumption of probabilistic independence of features leads to incorrect estimates

Fairness Verification with Graphical Models [GBM22a]

What did we achieve?

- We propose a method to include feature correlations using a Bayesian network leading to higher accuracy.
- A pseudo-polynomial fairness verification framework for *linear classifiers* that solves a stochastic subset-sum problem (S3P).

What did we lack?

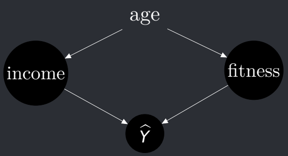
- Accuracy for Deep Nets: We do not have a formal and accurate verifier for nonlinear classifiers
- Scalability for Images and Texts: Bayesian network scales badly for high-dimensional data like images and texts.

Outline

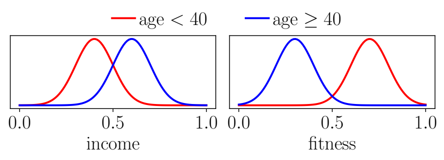
1. Motivation: Auditing, Understanding, and Eliminating Bias
2. Fairness Verification: Boolean Formulas with Independent Features
3. Fairness Explanation: A Model-agnostic Approach
4. Curtain Call: Question Avenir

Explaining Unfairness

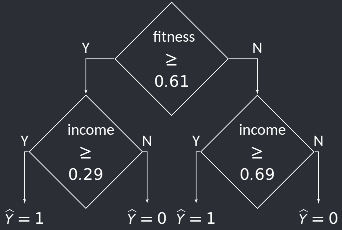
Data contains bias and classifiers trained on the data inherit the bias.



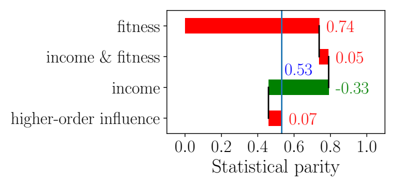
Dependency among features and prediction



Age-dependent distributions of non-sensitive features



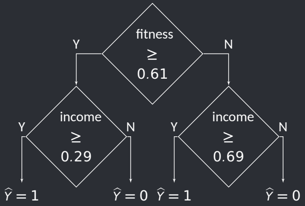
Decision tree (DT_1)



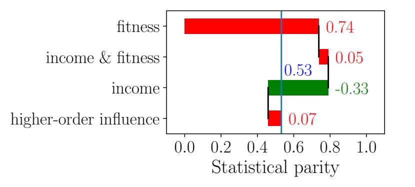
Fairness influence functions (FIF) of DT_1

Explaining Unfairness

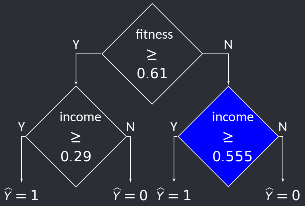
Identification of the source of unfairness is important to take affirmative actions.



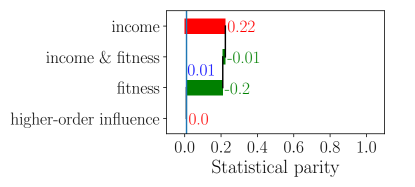
Decision tree (DT_1)



Fairness influence functions (FIF) for DT_1



Decision tree with affirmative action (DT_2)



Modified FIFs for DT_2

Fairness Influence Functions

A Model-agnostic Quantification of Fairness Explanations [GBM22b]

Fairness Influence Function (FIF) $w_{\mathbf{S}} : \mathbf{X}_{\mathbf{S}} \rightarrow \mathbb{R}$ measures the contribution of the *subset of features* $\mathbf{X}_{\mathbf{S}} \subseteq \mathbf{X}_{[k]}$ on the *bias* $f(\mathscr{A}, \mathbf{D})$ of the *classifier* $\mathscr{A}$ for *dataset* $\mathbf{D}$.

Axiom: Additivity of influence

Sum of FIFs of all subsets of non-sensitive features is equal to the bias of the classifier.

$$f(\mathscr{A}, \mathbf{D}) = \sum_{\mathbf{S} \subseteq [|x|] \setminus \emptyset} w_{\mathbf{S}}$$

FairXplainer: Computing Fairness Influence Functions

Key Ideas

1. Statistical parity is equal to a scaled difference between variance of outcomes for sensitive groups

If $p_{\mathrm{a}} \triangleq \max_{\mathrm{a}} \Pr[\hat{Y} = 1 | \mathrm{A} = \mathrm{a}]$ and $p_{\mathrm{a}'} \triangleq \min_{\mathrm{a}'} \Pr[\hat{Y} = 1 | \mathrm{A} = \mathrm{a}']$,

$$\text{Statistical Parity} = \frac{\mathrm{Var}[\hat{Y} = 1 | \mathrm{A} = \mathrm{a}] - \mathrm{Var}[\hat{Y} = 1 | \mathrm{A} = \mathrm{a}'])}{1 - (p_{\mathrm{a}} + p_{\mathrm{a}'})}$$

FairXplainer: Computing Fairness Influence Functions

Key Ideas

1. Statistical parity is equal to a scaled difference between variance of outcomes for sensitive groups
2. If we can decompose the variance in terms of the basis functions of the classifier, we can decompose the first and higher order variances as the variances of these decompositions.

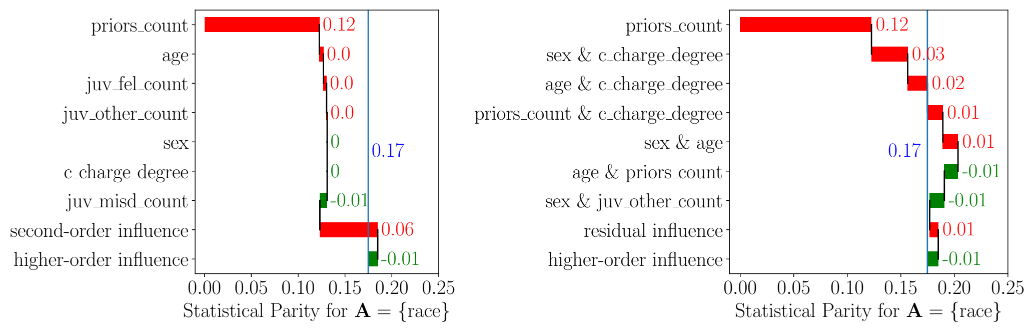
If $p_{\mathbf{a}} \triangleq \max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$ and $p_{\mathbf{a}'} \triangleq \min_{\mathbf{a}'} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}']$,

$$\text{Statistical Parity} = \frac{\mathrm{Var}[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] - \mathrm{Var}[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}'])}{1 - (p_{\mathbf{a}} + p_{\mathbf{a}'})}$$

$$= \frac{\sum_{i=1}^{n} \overbrace{(V_i^{(\mathbf{a})} - V_i^{(\mathbf{a}')})}^{1\text{-st order}} + \sum_{i<j}^{n} \overbrace{(V_{ij}^{(\mathbf{a})} - V_{ij}^{(\mathbf{a}')})}^{2\text{-nd order}} + \cdots + \overbrace{(V_{12\ldots n}^{(\mathbf{a})} - V_{12\ldots n}^{(\mathbf{a}')})}^{n\text{-th order}}}{1 - (p_{\mathbf{a}} + p_{\mathbf{a}'})}$$

Explaining Statistical Parity in COMPAS Dataset

Higher Order Effects are Decisive

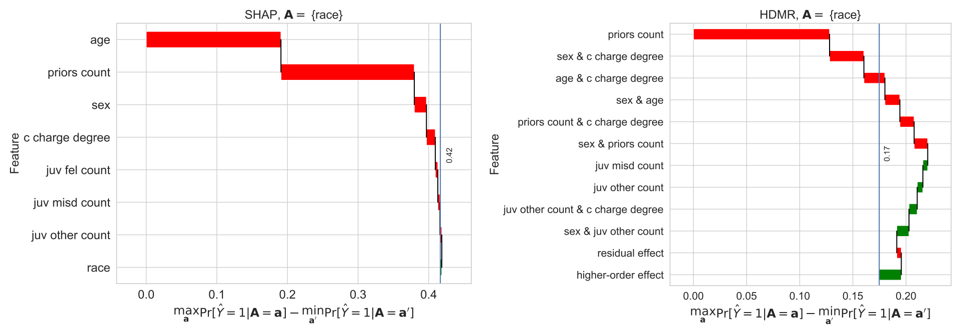


(a) FairXplainer: First order effects

(b) FairXplainer: First and second order effect

Explaining Statistical Parity in COMPAS Dataset

Local Explanations cannot Explain Unfairness

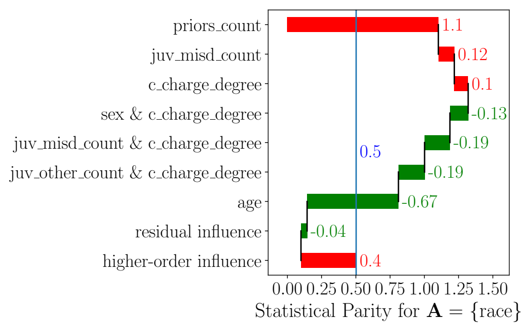


(c) Shapley Explanations

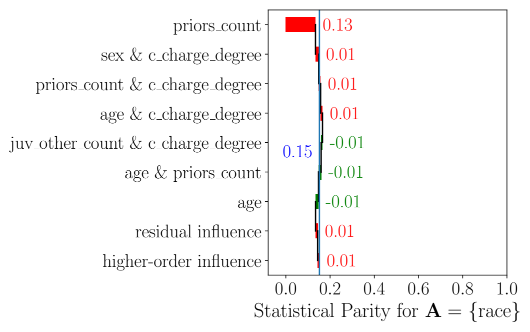
(d) FairXplainer: First and second order effect

Explaining Statistical Parity in COMPAS Dataset

FairXplainer can Detect Effects of Affirmative/Punitive Actions



(e) Fairness attack (punitive actions)



(f) Fairness enchancing algorithm (affirmative actions)

Summary of FairXplainer [GBM22b]

Axiomatic Formulation of Global Explanations

Observation: Fairness, particularly group fairness, is a global property of the classifier.

- We develop an axiomatic formulation of Fairness Influence Functions for any subset of features.

A Model-agnostic Algorithm

Observation: Fairness computation is equivalent to computing *the sensitivity of the classifier* w.r.t. different sensitive groups

- We propose FairXplainer that Extend global sensitivity analysis techniques from functional analysis to classification for computing FIFs

Outline

1. Motivation: Auditing, Understanding, and Eliminating Bias
2. Fairness Verification: Boolean Formulas with Independent Features
3. Fairness Explanation: A Model-agnostic Approach
4. Curtain Call: Question Avenir

Where are We?

- **Two facets of auditing bias of ML algorithms:**
  - **Fairness verification** allows us to macro-audit an ML algorithm
  - **Fairness explanation** allows us to detect sources of bias, and influences of affirmative/punitive actions

- **Fairness verifiers:** Justicia and FVGM improve scalability and accuracy for Boolean representable and linear classifiers

- **Fairness explanation:** FairXplainer identifies the effect of features and their interactions on the bias

What's ahead?

- Verification beyond Boolean and linear: Study scalable verifiers with formal guarantees for nonlinear classifiers, such as deep NN
- Elimination of bias: Using fairness verifiers and explainers to eliminate bias from ML algorithms

The Golden Goal: Regulating and Auditing Bias in ML

Developing and deploying theoretically-grounded standards for regulating and eliminating bias from digital data-dependent applications

Bibliography I

[ADDN17] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V Nori.
FairSquare: probabilistic verification of program fairness.
Proceedings of the ACM on Programming Languages, 1(OOPSLA):1–30, 2017.

[BDH+18] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang.
Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, Oct 2018.

[BZSL19] Osbert Bastani, Xin Zhang, and Armando Solar-Lezama.
Probabilistic verification of fairness properties via concentration.
Proceedings of the ACM on Programming Languages, 3(OOPSLA):1–27, 2019.

Bibliography II

[GBM21] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel.
Justicia: A stochastic SAT approach to formally verify fairness.
In *Proceedings of AAAI*, 2 2021.

[GBM22a] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel.
Algorithmic fairness verification with graphical models.
In *Proceedings of AAAI*, 2 2022.

[GBM22b] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel.
How biased is your feature?: Computing fairness influence functions with global sensitivity analysis.
2022.

[LWJ18] Nian-Ze Lee, Yen-Shi Wang, and Jie-Hong R Jiang.
Solving exist-random quantified stochastic boolean satisfiability via clause selection.
In *IJCAI*, pages 1339–1345, 2018.

Want to detect unfairness in your favourite classifier?

Use our Python library: “pip install justicia”

https://github.com/meelgroup/justicia

Joint works with Bishwamittra Ghosh and Kuldeep Meel, National Univ. of Singapore.

FairXplain: Key Ideas

Idea 1

Statistical parity can be computed using the difference between variance of outcomes for sensitive groups

If $p_{\mathbf{a}} \triangleq \max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$ and $p_{\mathbf{a}'} \triangleq \min_{\mathbf{a}'} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}']$,

$$\text{Statistical Parity} = \frac{\text{Var}[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] - \text{Var}[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}'])}{1 - (p_{\mathbf{a}} + p_{\mathbf{a}'})}$$

$$= \frac{\sum_{i=1}^{n} \overbrace{(V_i^{(\mathbf{a})} - V_i^{(\mathbf{a}')})}^{1\text{-st order}} + \sum_{i<j}^{n} \overbrace{(V_{ij}^{(\mathbf{a})} - V_{ij}^{(\mathbf{a}')})}^{2\text{-nd order}} + \cdots + \overbrace{(V_{12\ldots n}^{(\mathbf{a})} - V_{12\ldots n}^{(\mathbf{a}')})}^{n\text{-th order}}}{1 - (p_{\mathbf{a}} + p_{\mathbf{a}'})}$$

$$V_i^{(\mathbf{a})} = \text{Var}_{X_i}[\mathrm{E}_{X_{\sim i}}[\hat{Y} = 1 | X_i, \mathbf{A} = \mathbf{a}]], \quad V_{ij}^{(\mathbf{a})} = \text{Var}_{X_{ij}}[\mathrm{E}_{X_{\sim ij}}[\hat{Y} = 1 | X_i, X_j, \mathbf{A} = \mathbf{a}]] - V_i^{(\mathbf{a})} - V_j^{(\mathbf{a})}$$

FairXplain: Key Ideas

Idea 2

If we can decompose the variance in terms of the basis functions of the classifier, we can decompose the first and higher order variances as the variances of these decompositions.

$$f_{\{i\}}(\mathbf{X}_{\{i\}}) \approx \sum_{r=-1}^{m+1} \alpha_r^i B_r(\mathbf{X}_{\{i\}})$$

$$f_{\{i,j\}}(\mathbf{X}_{\{i,j\}}) \approx \sum_{p=-1}^{m+1} \sum_{q=-1}^{m+1} \beta_{pq}^{ij} B_p(\mathbf{X}_{\{i\}}) B_q(\mathbf{X}_{\{j\}})$$

$$f_{\{i,j,k\}}(\mathbf{X}_{\{i,j,k\}}) \approx \sum_{p=-1}^{m+1} \sum_{q=-1}^{m+1} \sum_{r=-1}^{m+1} \gamma_{pqr}^{ijk} B_p(\mathbf{X}_{\{i\}}) B_q(\mathbf{X}_{\{j\}}) B_r(\mathbf{X}_{\{j\}})$$

Fairness Verification with Graphical Models [GBM22a]

Justicia without Independence (Assumption): Accuracy

The schematic:

- Discretise each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
- Refine the CNF representation with the discretised features
- Learn a Bayesian network on the discretised features
- Run Justicia with the marginals from Bayesian network and the new CNF formula

Fairness Verification with Graphical Models [GBM22a]

Stochastic Subset Sum for Linear Classifiers: Scalability

The schematic:

- Discretise each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
- Refine the CNF representation with the discretised features
- Learn a Bayesian network on the discretised features
- Run Justicia with the marginals from Bayesian network and the new CNF formula