Verification and Explanation of Unfairness in Machine Learning

Debabrota Basu[a]

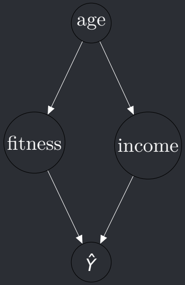
Joint work with Bishwamittra Ghosh and Kuldeep S. Meel[b]

[a] Équipe Scool, Univ. Lille, Inria, UMR 9189-CRIStAL, CNRS, Centrale Lille, France
[b] School of Computing, National University of Singapore, Singapore

(Un)Fairness in Machine Learning

Prediction of eligibility of health insurance

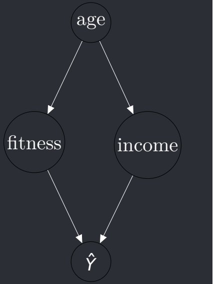
- Sensitive features, $\mathbf{A} = \{\text{age}\}$
- Non-sensitive features, $\mathbf{X} = \{\text{fitness, income}\}$

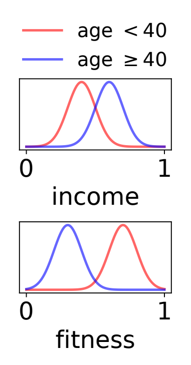


(Un)Fairness in Machine Learning

Prediction of eligibility of health insurance

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

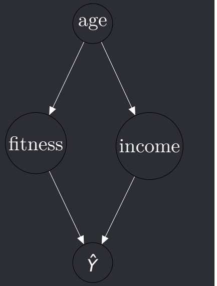


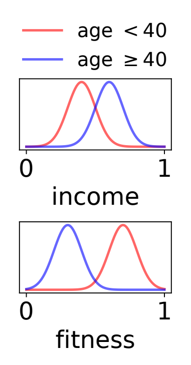


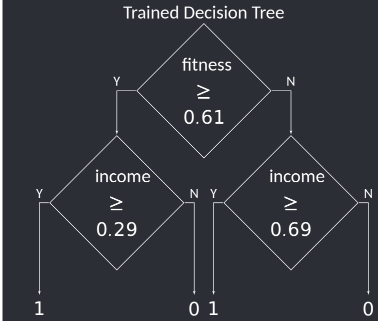
(Un)Fairness in Machine Learning

Prediction of eligibility of health insurance

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



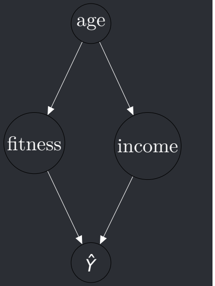


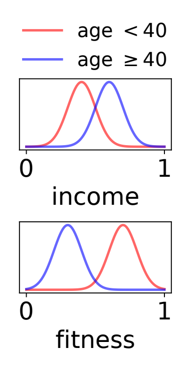


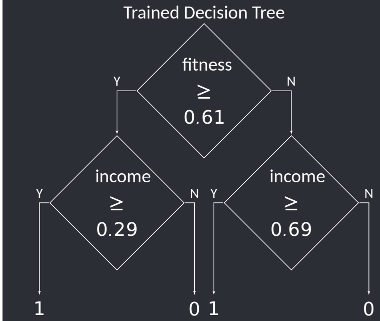
(Un)Fairness in Machine Learning

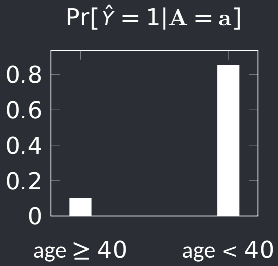
Prediction of eligibility of health insurance

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









Motivation

- Machine learning classifiers may become unfair to certain demographic groups

- Multiple fairness definitions and algorithms have been proposed to improve fairness

- What are still missing is scalable algorithms for verification and explanation of fairness

- Today, we focus on
  - **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
2. Fairness Verification of Boolean Formulas
3. Fairness Verification of Linear Classifiers with Correlated Features
4. Fairness Explanation: A Model-agnostic Approach

Justicia: A Stochastic SAT Approach to Formally Verify Fairness [1]

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. $\mathscr{D}$

Justicia: A Stochastic SAT Approach to Formally Verify Fairness [1]

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. $\mathscr{D}$

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

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

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

Justicia: A Stochastic SAT Approach to Formally Verify Fairness [1]

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. $\mathscr{D}$

$\Pr[\hat{Y} = 1|\mathbf{A} = \mathbf{a}]$ 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 ($ꓤ^p$)

Stochastic SAT (SSAT)

A Tale of Two Encodings

- **Existential-random SSAT formula**

$$\Phi_{\mathrm{ER}} = \exists x_2, \exists x_3, \rotatebox[origin=c]{180}{R}^{0.25} x_1,\ (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3) \land \neg x_1$$

- $\Pr[\Phi_{\mathrm{ER}}] = 0.75$
- Optimal assignment (maximization): $x_2 =$ false, $x_3 =$ false

Stochastic SAT (SSAT)

A Tale of Two Encodings

- **Existential-random SSAT formula**

$$\Phi_{\text{ER}} = \exists x_2, \exists x_3, \text{ꓤ}^{0.25} x_1,\ (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge \neg x_1$$

  - $\Pr[\Phi_{\text{ER}}] = 0.75$
  - Optimal assignment (maximization): $x_2 = \text{false}$, $x_3 = \text{false}$

- **Universal-random SSAT formula**

$$\Phi_{\text{UR}} = \forall x_2, \forall x_3, \text{ꓤ}^{0.25} x_1,\ (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge \neg x_1$$

  - $\Pr[\Phi_{\text{UR}}] = 0$
  - Optimal assignment (minimization): $x_2 = \text{true}$, $x_3 = \text{false}$

Justicia: Fairness Verification with SSAT

Consider

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

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

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

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

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

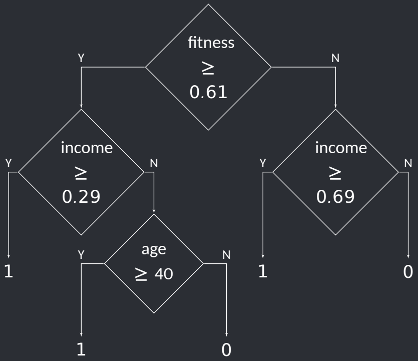
$$\Phi_{\mathrm{ER}} \triangleq \underbrace{\exists A_1, \ldots, \exists A_n,}_{\text{sensitive features}} \underbrace{\text{ꓤ}^{p_1} X_1, \ldots, \text{ꓤ}^{p_m} X_m,}_{\text{non-sensitive features}} \phi_{\hat{Y}}.$$

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

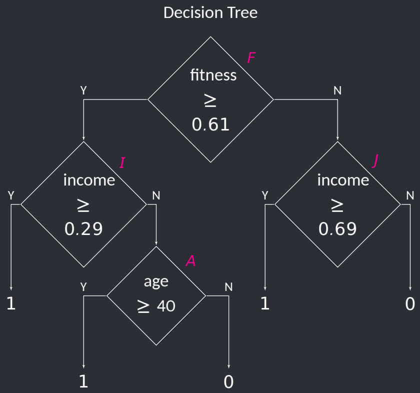
Use an SSAT solver to solve the ER-SSAT problems [2].

An Illustration

Decision Tree



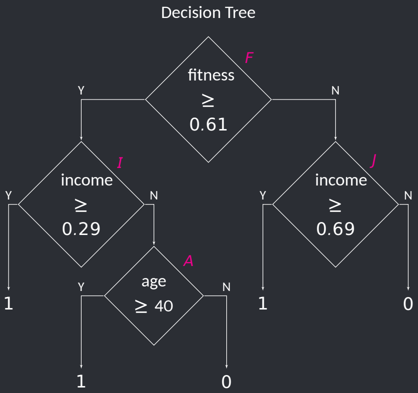
An Illustration



- 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

$$\Phi_{ER} = \exists A, \rotatebox[origin=c]{180}{R}^{0.41} F, \rotatebox[origin=c]{180}{R}^{0.93} I, \rotatebox[origin=c]{180}{R}^{0.09} J,\ (\neg F \vee I \vee A) \wedge (F \vee J)$$

An Illustration

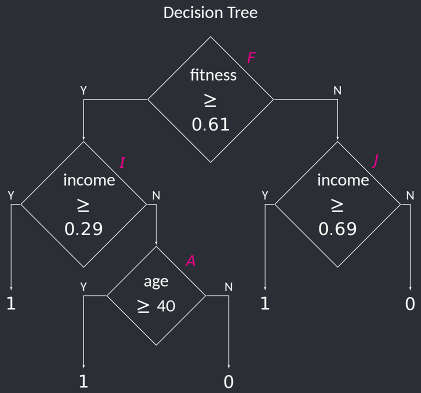

Decision Tree

- 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 $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, we construct

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

- $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] = \Pr[\Phi_{\mathrm{ER}}] = 0.46$

An Illustration



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

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

- $\max_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] = \Pr[\Phi_{\text{ER}}] = 0.46$
- Similarly, $\min_{\mathbf{a}} \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}] = 0.43$
- 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

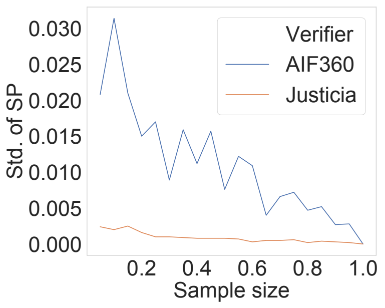


Figure: Robustness between probabilistic (Justicia) and dataset centric (AIF360 [3]) verifiers

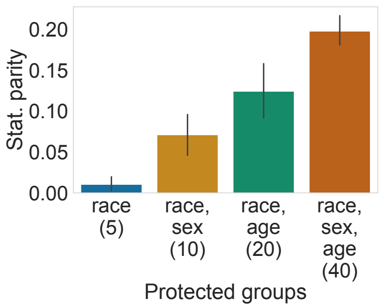


Figure: Verifying compound sensitive/protected groups with Justicia

Experimental Results

Faster than the Fastest

State-of-the-art probabilistic fairness verifiers

- FairSquare: computes weighted volume of logical programs using SMT reduction [4]
- VeriFair: probabilistic verification via sampling [5]

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

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

Outline

1. Motivation

2. Fairness Verification of Boolean Formulas

3. Fairness Verification of Linear Classifiers with Correlated Features

4. Fairness Explanation: A Model-agnostic Approach

FVGM: Algorithmic Fairness Verification with Graphical Models [6]

Fairness verification of Linear Classifiers

Challenges of earlier fairness verifiers

- **Scalability:** SSAT or SMT-based reduction of linear classifiers is computationally expensive
- **Accuracy:** Feature correlation is imprecisely modelled

FVGM: Algorithmic Fairness Verification with Graphical Models [6]

Fairness verification of Linear Classifiers

Challenges of earlier fairness verifiers

- **Scalability:** SSAT or SMT-based reduction of linear classifiers is computationally expensive
- **Accuracy:** Feature correlation is imprecisely modelled

Proposed solutions

- **Scalability:** Novel stochastic subset-sum problem (S3P) based reduction
- **Accuracy:** Feature correlations represented as a Bayesian network

Linear Classifiers

Let

- w_{X_i} be the the weight/coefficient of non-sensitive feature X_i
- w_{A_j}be the the weight/coefficient of sensitive feature A_j
- τ is the offset parameter

The prediction of a binary linear classifier

$$\hat{Y} = \mathbb{1}\Big[\sum_i w_{X_i} X_i + \sum_j w_{A_j} A_j \geq \tau\Big].$$

Linear Classifiers

Let

- w_{X_i} be the the weight/coefficient of non-sensitive feature X_i
- w_{A_j}be the the weight/coefficient of sensitive feature A_j
- τ is the offset parameter

The prediction of a binary linear classifier

$$\hat{Y} = \mathbb{1}\Big[\sum_i w_{X_i} X_i + \sum_j w_{A_j} A_j \geq \tau\Big].$$

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

by a reduction to S3P

A Detour to Subset-sum Problem

- $\mathbf{B} \triangleq \{B_i\}_{i=1}^{|\mathbf{B}|}$ be a set of Boolean variables
- $w_i \in \mathbb{Z}$ be the weight of B_i
- a constant threshold $\tau \in \mathbb{Z}$

Given a constraint

$$\sum_{i=1}^{|\mathbf{B}|} w_i B_i = \tau$$

the subset-sum problem computes $\mathbf{b} \in \{0, 1\}^{|\mathbf{B}|}$ such that the constraint evaluates to true when $\mathbf{B}$ is substituted with $\mathbf{b}$

Example:

- weights $\{-7, -3, -2, 9000, 5, 8\}$ and $\tau = 0$
- $\mathbf{b} = [0, 1, 1, 0, 1, 0]$ is the solution of the subset-sum problem, since $-3 - 2 + 5 = 0$

Stochastic Subset-sum Problem (S3P)

A Counting Analogue of the Subset-Sum Problem

S3P computes the *probability* of a subset of $\mathbf{B}$ with sum of weights of non-zero variables to be at least τ. Formally,

$$s(\mathbf{B}, \tau) \triangleq \Pr\left[\sum_i w_i B_i \geq \tau\right] \in [0, 1].$$

Stochastic Subset-sum Problem (S3P)

A Counting Analogue of the Subset-Sum Problem

S3P computes the *probability* of a subset of $\mathbf{B}$ with sum of weights of non-zero variables to be at least τ. Formally,

$$s(\mathbf{B}, \tau) \triangleq \Pr\left[\sum_i w_i B_i \geq \tau\right] \in [0, 1].$$

Similar to SSAT, we consider a quantifier $q_i \in \{ ꓤ^{p_i}, \exists, \forall \}$ for each B_i in S3P

Stochastic Subset-sum Problem (S3P)

The Semantics

Let $\mathrm{B}[2:n] \triangleq \{B_j\}_{j=2}^{n}$ be the subset of B without the first variable B_1.

$s(\mathrm{B}, \tau)$ is recursively defined as

$$s(\mathrm{B}, \tau) = \begin{cases} \mathbb{1}[\tau \leq 0], & \text{if } \mathbf{B} = \emptyset \\ s(\mathrm{B}[2:n], \tau - \max\{w_1, 0\}), & \text{if } q_1 = \exists \\ s(\mathrm{B}[2:n], \tau - \min\{w_1, 0\}), & \text{if } q_1 = \forall \\ p_1 \times s(\mathrm{B}[2:n], \tau - w_1) + (1 - p_1) \times s(\mathrm{B}[2:n], \tau), & \text{if } q_1 = \text{Я}^{p_1} \end{cases}$$

Stochastic Subset-sum Problem (S3P)

Differences of S3P *with SSAT*

- Computation of $\exists$ and $\forall$ quantified variables is linear in S3P but exponential in SSAT.

- There is a pseudo-polynomial dynamic programming algorithm for S3P compared to the $\mathrm{NP}^{\mathrm{PP}}$-hardness of ER-SSAT and UR-SSAT.

FVGM: Fairness Verification of Linear Classifiers

1. Preprocess a linear classifier
    - discretize each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
    - if w is the weight of X and μ_i is the mean of feature values in the i-th bin, then the weight of $B_i \in \mathbf{B}$ is $w\mu_i$

FVGM: Fairness Verification of Linear Classifiers

1. Preprocess a linear classifier
   - discretize each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
   - if w is the weight of X and μ_i is the mean of feature values in the i-th bin, then the weight of $B_i \in \mathbf{B}$ is $w\mu_i$

2. Learn a Bayesian network on discretized features[1]

FVGM: Fairness Verification of Linear Classifiers

1. Preprocess a linear classifier
   - discretize each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
   - if w is the weight of X and μ_i is the mean of feature values in the i-th bin, then the weight of $B_i \in \mathbf{B}$ is $w\mu_i$

2. Learn a Bayesian network on discretized features[1]

3. To compute $\max_a \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$,
   - assign $\exists$ quantifier to sensitive features $\mathbf{A}$
   - assign $\reflectbox{R}$ quantifier to non-sensitive features $\mathbf{X}$
   - solve S3P problem

FVGM: Fairness Verification of Linear Classifiers

1. Preprocess a linear classifier
   - discretize each continuous feature X to a set of Boolean features $\mathbf{B}$ using histogram
   - if w is the weight of X and μ_i is the mean of feature values in the i-th bin, then the weight of $B_i \in \mathbf{B}$ is $w\mu_i$

2. Learn a Bayesian network on discretized features[1]

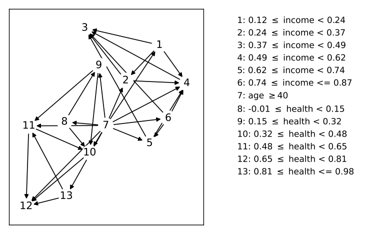
3. To compute $\max_a \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$,
   - assign ∃ quantifier to sensitive features $\mathbf{A}$
   - assign Я quantifier to non-sensitive features $\mathbf{X}$
   - solve S3P problem

4. To compute $\min_a \Pr[\hat{Y} = 1 | \mathbf{A} = \mathbf{a}]$, assign ∀ quantifier to $\mathbf{A}$ while keeping Я quantifier on $\mathbf{X}$

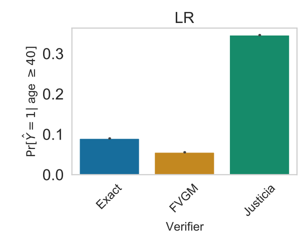
Experimental Analysis

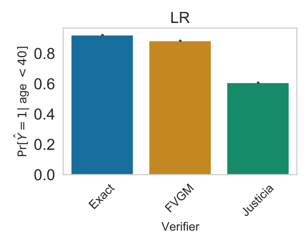
Accuracy

- Sensitive features, $\mathbf{A} = \{\text{age}\}$
- Non-sensitive features, $\mathbf{X} = \{\text{health, income}\}$
- We discretize $\mathbf{X}$ to Boolean features



1: 0.12 ≤ income < 0.24
2: 0.24 ≤ income < 0.37
3: 0.37 ≤ income < 0.49
4: 0.49 ≤ income < 0.62
5: 0.62 ≤ income < 0.74
6: 0.74 ≤ income <= 0.87
7: age ≥ 40
8: -0.01 ≤ health < 0.15
9: 0.15 ≤ health < 0.32
10: 0.32 ≤ health < 0.48
11: 0.48 ≤ health < 0.65
12: 0.65 ≤ health < 0.81
13: 0.81 ≤ health <= 0.98





Experimental Analysis

Scalability

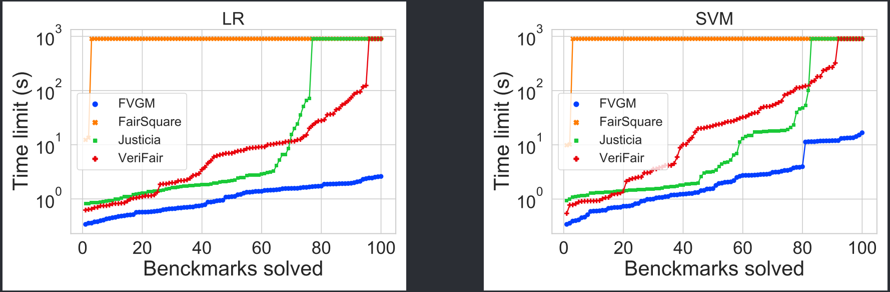


Figure: A cactus plot to present the scalability of different fairness verifiers on Linear Regression (LR) classifiers and Support Vector Machine (SVM)

Summary of FVGM

- FVGM is an efficient fairness verification framework for linear classifiers based on a novel stochastic subset-sum problem (S3P).
- FVGM is the first method to include feature correlations using a Bayesian network.
- FVGM demonstrates higher *scalability* and higher *accuracy* in comparison with earlier fairness verifiers.

Outline

1. Motivation

2. Fairness Verification of Boolean Formulas

3. Fairness Verification of Linear Classifiers with Correlated Features

4. Fairness Explanation: A Model-agnostic Approach

Fairness Explanation

- Identification of the source of unfairness is important to take affirmative actions
- Data contains bias and classifiers trained on the data inherit the bias.

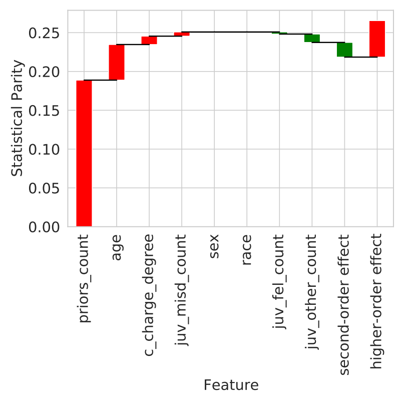


Figure: Explaining statistical parity in COMPAS recidivism prediction dataset for the feature ’sex’

Computing the Fairness Explanations

A Model-agnostic Approach

Observations

- Fairness, particularly group fairness, is a global property of the classifier.
- Fairness computation is equivalent to computing *the sensitivity of the classifier* w.r.t. different sensitive groups

Our approach: Extend global sensitivity analysis techniques from functional analysis to classification for explainning fairness.

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

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

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

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

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.

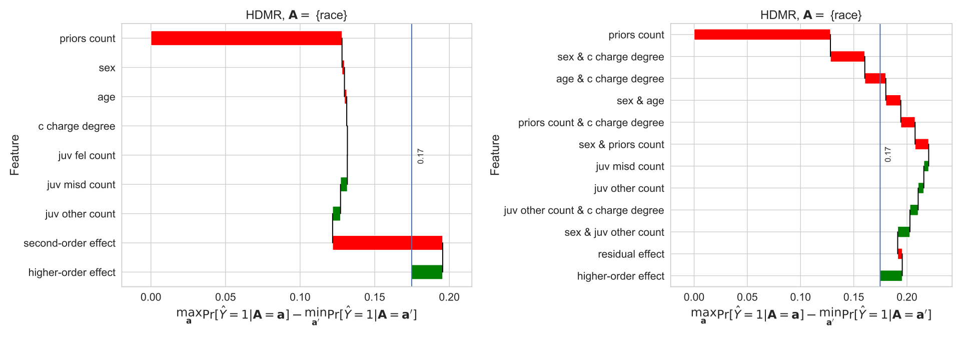
$$f_{\{i\}}(\mathbf{X}_{\{i\}}) \approx \sum_{r=-1}^{m+1} \alpha_r^i B_r(\mathbf{X}_{\{i\}})$$

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

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

Explaining Statistical Parity in COMPAS Dataset

Higher Order Effects are Decisive

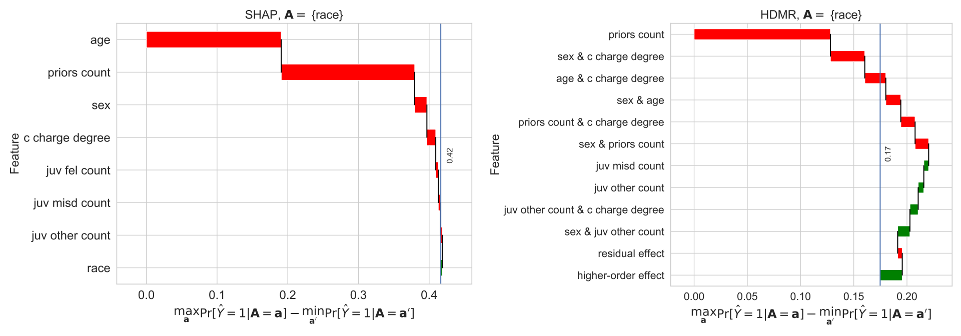


(a) FairXplain: First order effects

(b) FairXplain: First and second order effect

Explaining Statistical Parity in COMPAS Dataset

Local Explanations cannot Explain Unfairness



(c) Shapley Explanations

(d) FairXplain: First and second order effect

Conclusion

- **Fairness verification** and **explanation** are important problems in estimating the bias of classifiers and identifying the source of bias

- **Fairness verifiers**, Justicia and FVGM, improve upon existing fairness verifiers in terms of scalability and accuracy

- **Fairness explanation** shows the potential in identifying the effect of individual features or their interactions on the unfairness of the classifier. We currently focus in it.

- As a future work, we aim to design fairness enhancing algorithms relying on fairness verification and explanation

Bibliography I

[1] B. Ghosh, D. Basu, and K. S. Meel, “Justicia: A stochastic SAT approach to formally verify fairness,” in *Proceedings of AAAI*, 2 2021.

[2] N.-Z. Lee, Y.-S. Wang, and J.-H. R. Jiang, “Solving exist-random quantified stochastic boolean satisfiability via clause selection.” in *IJCAI*, 2018, pp. 1339–1345.

[3] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, “Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias,” Oct 2018. [Online]. Available: https://arxiv.org/abs/1810.01943

Bibliography II

[4] A. Albarghouthi, L. D'Antoni, S. Drews, and A. V. Nori, "FairSquare: probabilistic verification of program fairness," *Proceedings of the ACM on Programming Languages*, vol. 1, no. OOPSLA, pp. 1–30, 2017.

[5] O. Bastani, X. Zhang, and A. Solar-Lezama, "Probabilistic verification of fairness properties via concentration," *Proceedings of the ACM on Programming Languages*, vol. 3, no. OOPSLA, pp. 1–27, 2019.

[6] B. Ghosh, D. Basu, and K. S. Meel, "Algorithmic fairness verification with graphical models," in *Proceedings of AAAI*, 2 2022.

Want to detect unfairness in your favourite classifier?

Use our Python library: “pip install justicia”

https://github.com/meelgroup/justicia