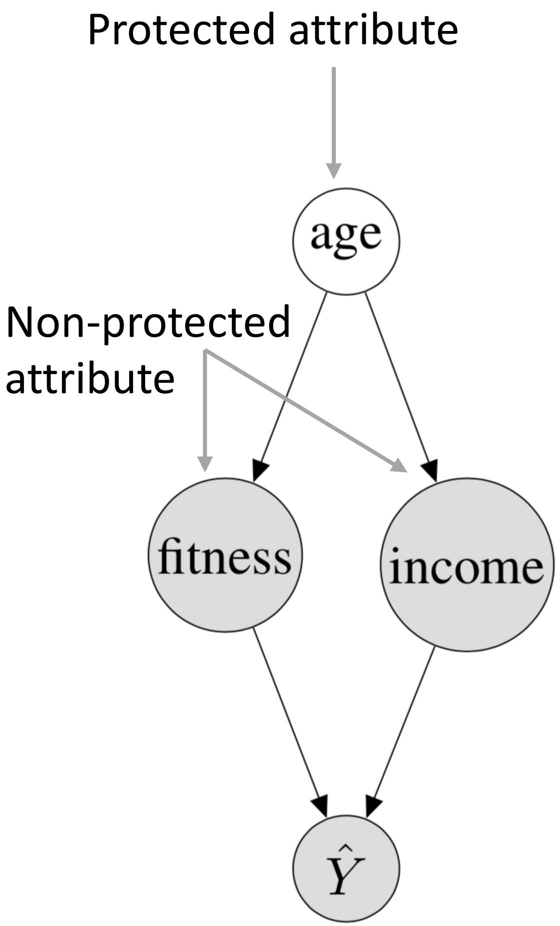
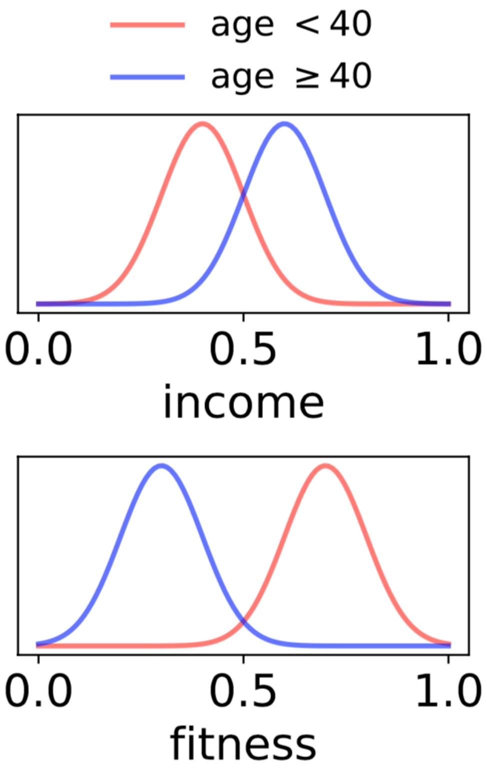
Justicia: A Stochastic SAT Approach to Formally Verify Fairness

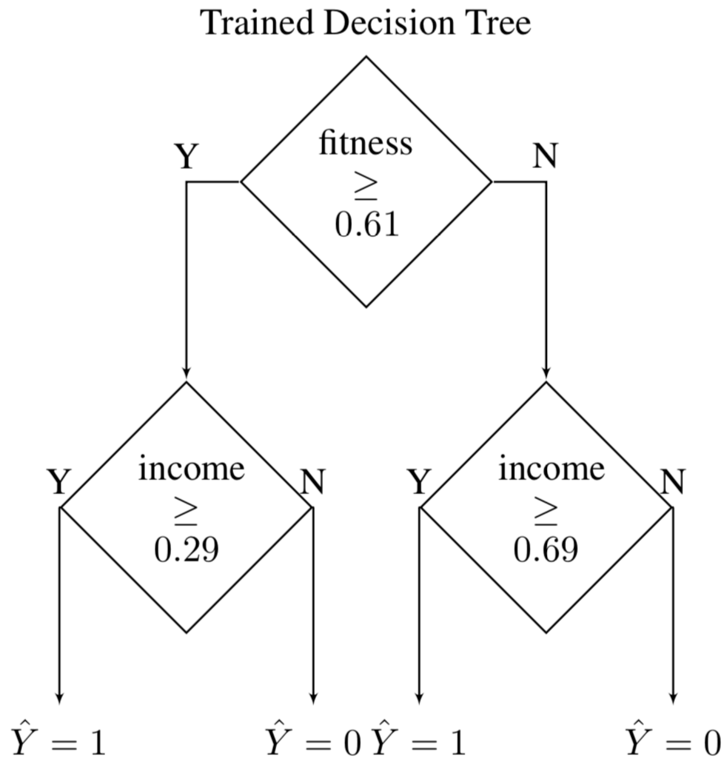
Bishwamittra Ghosh[1], Debabrota Basu[2,3], Kuldeep S. Meel[1]

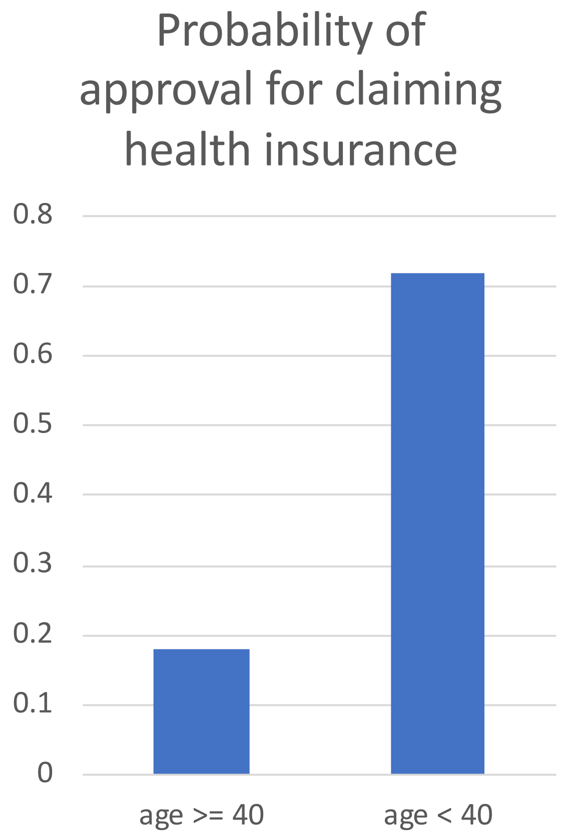
[1]National University of Singapore, Singapore
[2]Chalmers University of Technology, Goteborg, Sweden
[3]Scool, Inria Lille-Nord Europe, France

Unfairness in machine learning









Motivation

Fairness metrics

- Independence
  - Disparate impact
  - Statistical parity
- Separation
  - Equalized odds
- Sufficiency
  - Causal fairness

Fairness algorithms

- Preprocessing
- In-processing
- Postprocessing

A framework for verifying different fairness metrics and algorithms

Contribution

Fairness verification framework Justicia based on Stochastic SAT (SSAT)

- Two fairness definitions: independence and separation
- Handle compound protected groups
  - White-male, Black-female etc.
- Scalable
- Robust

Problem statement

- X = non-protected attributes
- A = protected attributes
- Y = true class label, $\hat{Y}$ = predicted class label

Given

- binary classifier $\mathcal{M} : (X, A) \rightarrow \{0,1\}$
- probability distribution $X \sim \mathcal{D}$

verify whether $\mathcal{M}$ achieves independence and separation metrics with respect to the distribution $\mathcal{D}$

Key observation

Computing positive predictive value (PPV)

$$\Pr\left[\hat{Y} = 1 \middle| A = \boldsymbol{a}\right]$$

is the building block of different fairness metrics

Two approaches

- Approach 1: enumeration on each $A = \boldsymbol{a}$
- Approach 2: learning most favored group $\boldsymbol{a}_{\text{fav}}$ and least favored group $\boldsymbol{a}_{\text{unfav}}$ based on PPV

Stochastic SAT (SSAT)

An SSAT formula has a prefix and a CNF formula ϕ

$$\Phi = \underbrace{Q_1 X_1, \ldots, Q_m X_m}_{\text{prefix}}, \phi$$

Q_i is either

- universal (∀),
- existential (∃), or
- randomized (R^{p_i}) quantification with $p_i = \Pr[X_i = \mathrm{TRUE}]$

The goal in SSAT is to compute the probability of satisfaction $\Pr[\Phi]$

Example of SSAT

$$\Phi = \mathrm{R}^{0.25}X_1, \exists X_2, \exists X_3,\ (X_1 \vee \neg X_2) \wedge (\neg X_1 \vee X_2 \vee X_3) \wedge (\neg X_1)$$

Semantics of SSAT

1. $\Pr[\mathrm{TRUE}] = 1, \Pr[\mathrm{FALSE}] = 0,$
2. $\Pr[\Phi] = \max_{X}\{\Pr[\Phi|_X], \Pr[\Phi|_{\neg X}]\}$ if X is existentially $(\exists)$ quantified
3. $\Pr[\Phi] = \min_{\mathrm{X}}\{\Pr[\Phi|_X], \Pr[\Phi|_{\neg X}]\}$ if X is universally $(\forall)$ quantified
4. $\Pr[\Phi] = p\Pr[\Phi|_X]$ + (1-p)$\Pr[\Phi|_{\neg X}]$ if X is randomized (R^{p_i}) quantified

where $\Phi|_X$ is the substitution of left-most variable in the prefix with $X = \mathrm{TRUE}$

Solution from an SSAT solver: $\Pr[\Phi] = 0.75$

Approach 1: Enumeration encoding

Consider a simple case

- Attributes $X \cup A$ are Boolean
- Classifier $\hat{Y}$ is a CNF formula $\phi_{\hat{Y}}$
- $p_i = \Pr[X_i]$ is known for each non-protected attribute

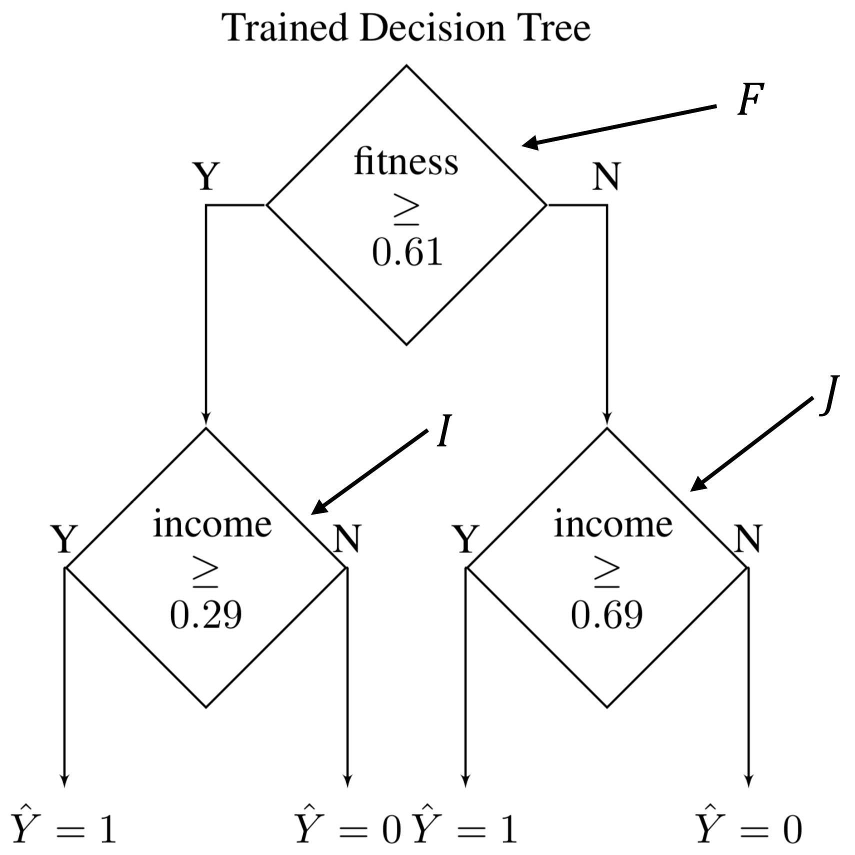
The computation of

$$\Pr[\hat{Y} = 1 | A = \boldsymbol{a}]$$

is equivalent to solving

$$\Phi_{\boldsymbol{a}} := \boxed{\underbrace{\mathrm{R}^{p_1} X_1, \ldots, \mathrm{R}^{p_m} X_m}_{\text{non–protected}}, \underbrace{\exists A_1, \ldots, \exists A_n}_{\text{protected}},} \boxed{\phi_{\hat{Y}} \wedge (A = \boldsymbol{a})}$$

Example of enumeration encoding



- Classifier $\phi_{\hat{Y}} = (\neg F \lor I) \land (F \lor J)$
- Let literal A = age ≥ 40 and $\neg A$ = age < 40

SSAT formula for "age $\geq$ 40" group:

$$\Phi_{\text{age}\geq 40} = \mathrm{R}^{0.41}F, \mathrm{R}^{0.93}I, \mathrm{R}^{0.09}J, \exists A, (\neg F \lor I) \land (F \lor J) \land A$$

Solving, $\Pr[\Phi_{\text{age}\geq 40}] = 0.43$

SSAT formula for "age $<$ 40" group:

$$\Phi_{\text{age}<40} = \mathrm{R}^{0.41}F, \mathrm{R}^{0.93}I, \mathrm{R}^{0.09}J, \exists A, (\neg F \lor I) \land (F \lor J) \land \neg A$$

Similarly, $\Pr[\Phi_{\text{age}<40}] = 0.43$

Computation of fairness metrics

- Disparate impact:

$$\frac{\Pr[\hat{Y}=1|\text{age}\geq 40]}{\Pr[\hat{Y}=1|\text{age}<40]} = \frac{0.43}{0.43} = 1$$

- Statistical parity:

$$|\Pr[\hat{Y} = 1|\text{age} \geq 40] - \Pr[\hat{Y} = 1|\text{age} < 40]| = |0.43 - 0.43| = 0$$

It looks like there is no discrimination

We did not consider correlation among attributes

Enumeration encoding with correlation

Use $\Pr[F|\text{age} \geq 40]$ instead of $\Pr[F]$ ...

$$\Phi_{\text{age}\geq 40} = \mathrm{R}^{0.01}F, \mathrm{R}^{0.99}I, \mathrm{R}^{0.18}J, \exists A, (\neg F \vee I) \wedge (F \vee J) \wedge A$$

With correlation, $\Pr\left[\hat{Y} = 1 \middle| \text{age} \geq 40\right] = 0.18$
Similarly, $\Pr\left[\hat{Y} = 1 \middle| \text{age} < 40\right] = 0.72$

Disparate impact $= \frac{0.18}{0.72} \neq 1$
Statistical parity $= 0.72 - 0.18 = 0.54 \neq 0$

Appraoch 2: Learning encoding

- Enumeration encoding has to be solved for exponential combinations of compound protected groups
- SSAT allows us to learn the assignment to existential (∃) and universal (∀) variables
- Learning the most favored group

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

- Learning the least favored group

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

Experiments

- State of the art
  - FairSquare: computes weighted volume of logical program using SMT
  - VeriFair: probabilistic verification via sampling
  - AIF360 (computes metrics on a finite dataset)

- Classifiers:
  - Linear classifier (pseudo-Boolean encoding)
  - Decision tree

Accuracy

Metric	Exact	Justicia	FairSquare	VeriFair	AIF360
Disparate impact	0.26	0.25	0.99	0.99	0.25
Stat. parity	0.53	0.54	—	—	0.54

Justicia has less than 1%-error

Scalability

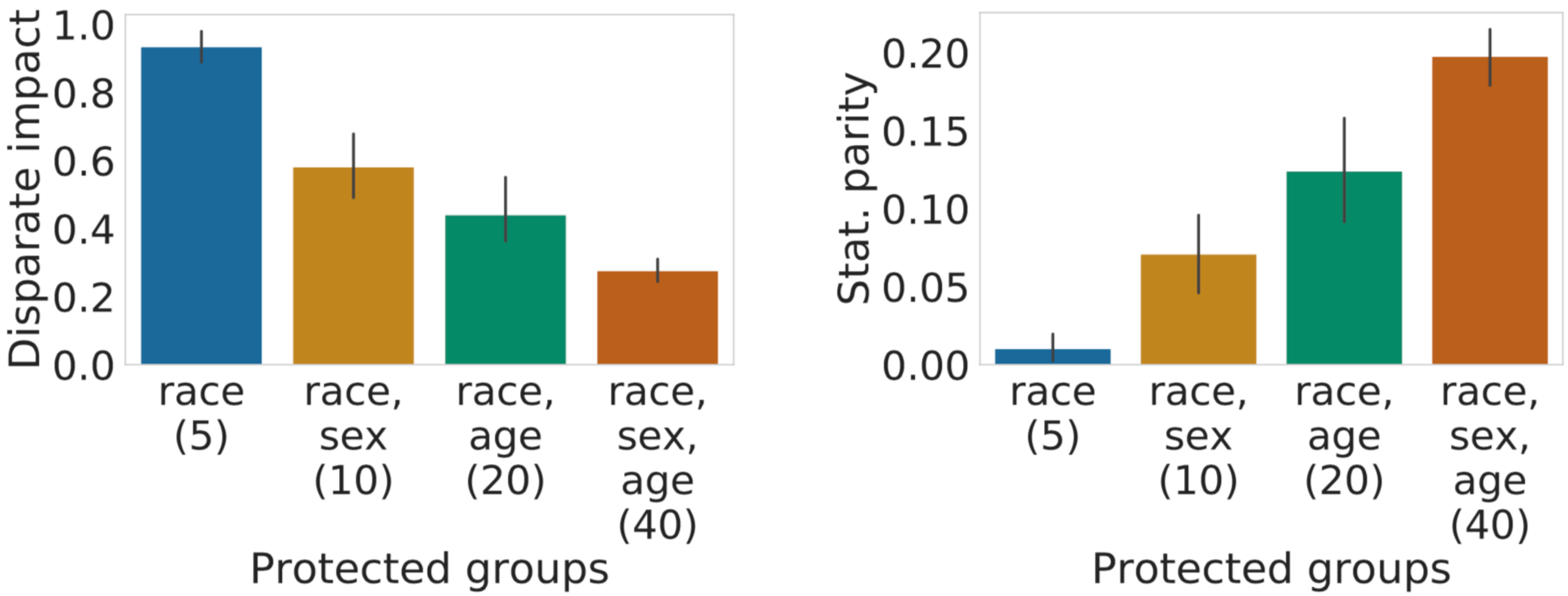
Dataset	Ricci		Titanic		COMPAS		Adult	
Classifier	DT	LR	DT	LR	DT	LR	DT	LR
Justicia	0.1	0.2	0.1	0.9	0.1	0.2	0.2	1.0
FairSquare	4.8	—	16.0	—	36.9	—	—	—
VeriFair	5.3	2.2	1.2	0.8	15.9	11.3	295.6	61.1

DT = decision tree

LR = logistic regression classifier

Justicia reports 1 to 3 orders of magnitude speed-up

Compound protected groups



Conclusion

- A stochastic SAT-based approach to formally verify different fairness metrics and algorithms
- First method to verify compound protected groups
- More accurate, scalable and robust than state-of-the-art methods

- Python library: pip install justicia

https://github.com/meelgroup/justicia

Source code & paper