Justicia: A Stochastic SAT Approach to Formally Verify Fairness

Bishwamittra Ghosh¹, Debabrota Basu²,³, Kuldeep S. Meel¹

¹National University of Singapore, Singapore
²Chalmers University of Technology, Goteborg, Sweden
³Scool, Inria Lille-Nord Europe, France
Unfairness in machine learning

Protected attribute

Non-protected attribute

age

fitness

income

\( \hat{Y} \)

Trained Decision Tree

\( \text{fitness} \geq 0.61 \)

\( \text{income} \geq 0.29 \)

\( \text{income} \geq 0.69 \)

Probability of approval for claiming health insurance

\begin{align*}
\text{age} < 40 & : 0.0 \quad \text{age} \geq 40 & : 0.8
\end{align*}
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

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

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)

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

is the building block of different fairness metrics

Two approaches

• Approach 1: enumeration on each \( A = a \)
• Approach 2: learning most favored group \( a_{fav} \) and least favored group \( a_{unfav} \) based on PPV
Stochastic SAT (SSAT)

An SSAT formula has a prefix and a CNF formula $\phi$

$$\Phi = Q_1X_1, \ldots, Q_mX_m, \phi$$

$Q_i$ is either
- universal ($\forall$),
- existential ($\exists$), or
- randomized ($R^{p_i}$) quantification with $p_i = Pr[X_i = \text{TRUE}]$

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

$$\Phi = R^{0.25}X_1, \exists X_2, \exists X_3, (X_1 \lor \neg X_2) \land (\neg X_1 \lor X_2 \lor X_3) \land (\neg X_1)$$

Semantics of SSAT
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 (\(\exists\)) quantified
3. \(\Pr[\Phi] = \min_{X} \{\Pr[\Phi|_X], \Pr[\Phi|_{\neg X}]\}\) 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 \(X = \text{TRUE}\)

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

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

is equivalent to solving

$$\Phi_a := R^{p_1}X_1, \ldots, R^{p_m}X_m, \exists A_1, \ldots, \exists A_n, \phi_{\hat{Y}} \land (A = a)$$

non-protected  protected
Example of enumeration encoding

- Classifier $\phi = (\neg F \lor I) \land (F \lor J)$
- Let literal $A = \text{age} \geq 40$ and $\neg A = \text{age} < 40$

SSAT formula for “age $\geq 40$” group:

$$\Phi_{age\geq40} = R^{0.41}F, R^{0.93}I, R^{0.09}J, \exists A, (\neg F \lor I) \land (F \lor J) \land A$$

Solving, $\Pr[\Phi_{age\geq40}] = 0.43$

SSAT formula for “age $< 40$” group:

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

Similarly, $\Pr[\Phi_{age<40}] = 0.43$
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 $\text{Pr}[F\mid \text{age} \geq 40]$ instead of $\text{Pr}[F]$ ...

\[ \Phi_{\text{age} \geq 40} = R^{0.01}F, R^{0.99}I, R^{0.18}J, \exists A, (\neg F \lor I) \land (F \lor J) \land A \]

With correlation, $\text{Pr}[\hat{Y} = 1\mid \text{age} \geq 40] = 0.18$

Similarly, $\text{Pr}[\hat{Y} = 1\mid \text{age} < 40] = 0.72$

Disparate impact $= \frac{0.18}{0.72} \neq 1$

Statistical parity $= 0.72 - 0.18 = 0.54 \neq 0$
Approach 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 ($\exists$) and universal ($\forall$) variables
• Learning the most favored group

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

• Learning the least favored group

$$\Phi_{\text{unfav}} = \forall A, R^{0.41} F, R^{0.93} I, R^{0.09} J, (\neg F \lor I) \land (F \lor J)$$
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

<table>
<thead>
<tr>
<th>Metric</th>
<th>Exact</th>
<th>Justicia</th>
<th>FairSquare</th>
<th>VeriFair</th>
<th>AIF360</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disparate impact</td>
<td>0.26</td>
<td>0.25</td>
<td>0.99</td>
<td>0.99</td>
<td>0.25</td>
</tr>
<tr>
<td>Stat. parity</td>
<td>0.53</td>
<td>0.54</td>
<td>—</td>
<td>—</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Justicia has less than 1%-error
## Scalability

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ricci</th>
<th></th>
<th></th>
<th>COMPAS</th>
<th></th>
<th></th>
<th></th>
<th>Adult</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DT</td>
<td>LR</td>
<td>DT</td>
<td>LR</td>
<td>DT</td>
<td>LR</td>
<td></td>
<td>DT</td>
<td>LR</td>
<td></td>
</tr>
<tr>
<td>Justicia</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td>0.2</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>FairSquare</td>
<td>4.8</td>
<td>—</td>
<td>16.0</td>
<td>—</td>
<td>36.9</td>
<td>—</td>
<td></td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>VeriFair</td>
<td>5.3</td>
<td>2.2</td>
<td>1.2</td>
<td>0.8</td>
<td>15.9</td>
<td>11.3</td>
<td></td>
<td>295.6</td>
<td>61.1</td>
<td></td>
</tr>
</tbody>
</table>

DT = decision tree  
LR = logistic regression classifier  

Justicia reports 1 to 3 orders of magnitude speed-up
Compound protected groups

Disparate impact

- race (5)
- race, sex (10)
- race, age (20)
- race, sex, age (40)

Stat. parity

- race (5)
- race, sex (10)
- race, age (20)
- race, sex, age (40)
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