FairTest:
Discovering unwarranted associations in data-driven applications

IEEE EuroS&P
April 28th, 2017

Florian Tramèr[1], Vaggelis Atlidakis[2], Roxana Geambasu[2], Daniel Hsu[2], Jean-Pierre Hubaux[3], Mathias Humbert[4], Ari Juels[5], Huang Lin[3]

[1]Stanford University, [2]Columbia University, [3]École Polytechnique Fédérale de Lausanne, [4]Saarland University, [5]Cornell Tech

“Unfair” associations + consequences

Websites Vary Prices, Deals Based on Users' Information

By JENNIFER VALENTINO-DEVRIES, JEREMY SINGER-VINE and ASHKAN SOLTANI

December 24, 2012

It was the same Swingline stapler, on the same Staples.com website. But for Kim Wamble, the price was $15.79, while the price on Trude Frizzell's screen, just a few miles away, was $14.29.

A key difference: where Staples seemed to think they were located.

In what appears to be an unintended side effect of Staples' pricing methods—likely a function of retail competition with its rivals—the Journal's testing also showed that areas that tended to see the discounted prices had a higher average income than areas that tended to see higher prices.

“Unfair” associations + consequences

Google Photos labeled black people 'gorillas'

Jessica Guynn, USA TODAY 2:10 p.m. EDT July 1, 2015

SAN FRANCISCO — Google has apologized after its new Photos application identified black people as "gorillas."

On Sunday Brooklyn programmer Jacky Alciné tweeted a screenshot of photos he had uploaded in which the app had labeled Alcine and a friend, both African American, "gorillas."

Yontan Zunger, an engineer and the company's chief architect of Google+, responded swiftly to Alciné on Twitter: "This is 100% Not OK." And he promised that Google's Photos team was working on a fix.

These are **software bugs**: need to *actively test for them* and *fix them (i.e., debug)* in data-driven applications…
just as with functionality, performance, and reliability bugs.

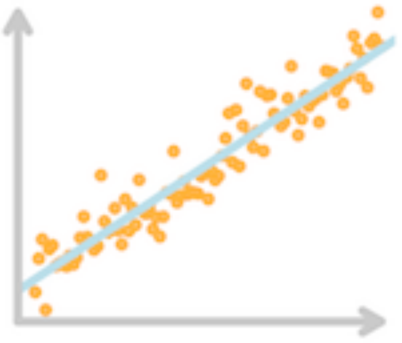
Unwarranted Associations Model

User inputs

Data-driven application

Application outputs

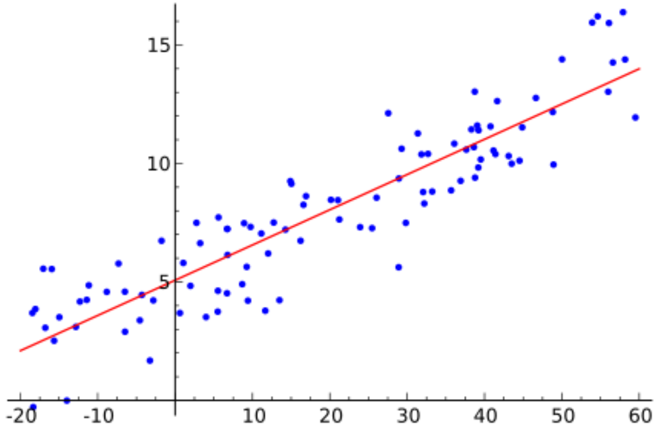
Protected inputs

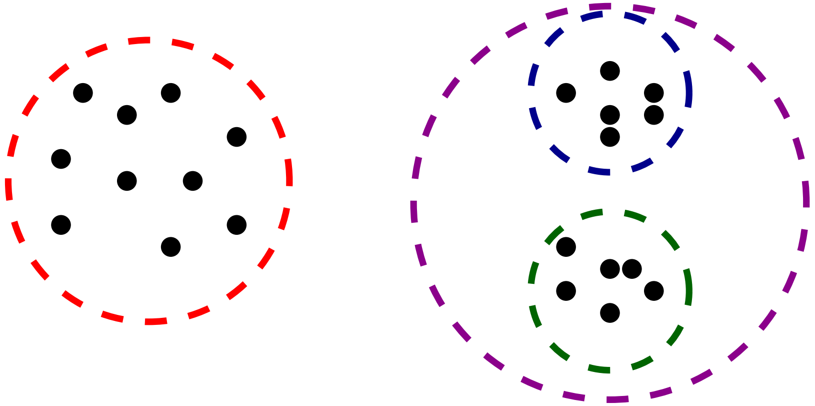


Limits of preventative measures

What doesn't work:

- <u>Hide protected attributes</u> from data-driven application.
- Aim for <u>statistical parity</u> w.r.t. protected classes and service output.





Foremost challenge is to even detect these unwarranted associations.

A Framework for Unwarranted Associations

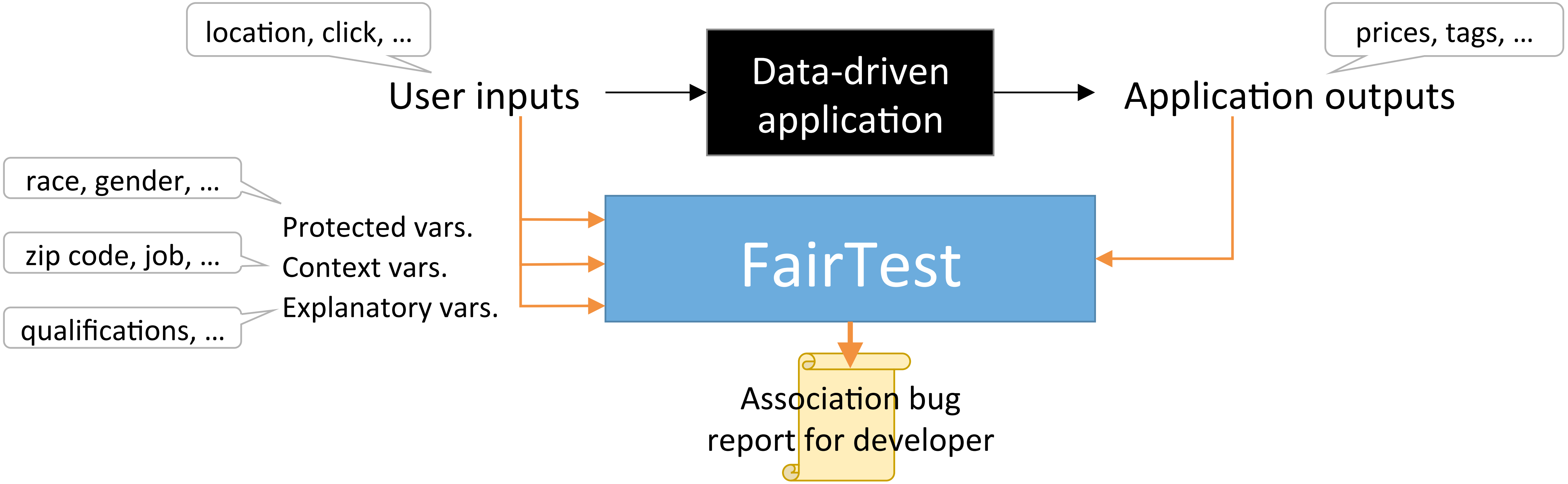
1. Specify **relevant data features**:
   - Protected variables (e.g., Gender, Race, …)
   - "Utility": a function of the algorithm's output (e.g., Price, Error rate, …)
   - Explanatory variables (e.g., Qualifications)
   - Contextual variables (e.g., Location, Job, …)

2. Find **statistically significant associations** between protected attributes and utility
   - *Condition* on explanatory variables
   - Not tied to any particular *statistical metric* (e.g., odds ratio)

3. Granular search in **semantically meaningful subpopulations**
   - Efficiently list subgroups with highest adverse effects

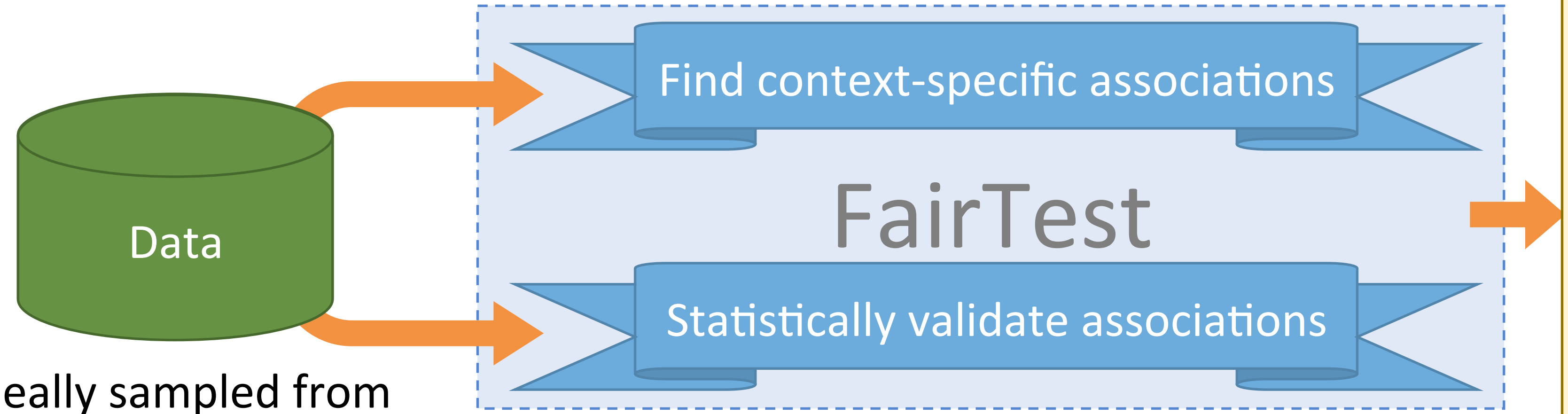
FairTest: a testing suite for data-driven apps

- Finds context-specific associations between protected variables and application outputs, conditioned on explanatory variables
- Bug report ranks findings by assoc. strength and affected pop. size



A data-driven approach

Core of FairTest is based on statistical machine learning



Ideally sampled from relevant user population

Statistical machine learning internals:

- top-down spatial partitioning algorithm
- confidence intervals for assoc. metrics
- …

Report of associations of O=Price on S_i=Income:
Assoc. metric: norm. mutual information (NMI).

Global Population of size 494,436
p-value=3.34e-10 ; NMI=[0.0001, 0.0005]

Price	Income <$50K	Income >=$50K	Total
High	**15301 (6%)**	**13867 (6%)**	29168 (6%)
Low	234167 (94%)	231101 (94%)	465268 (94%)
Total	249468 (50%)	244968 (50%)	494436 (100%)

1. Subpopulation of size 23,532
Context={State: CA, Race: White}
p-value=2.31e-24 ; NMI=[0.0051, 0.0203]

Price	Income <$50K	Income >=$50K	Total
High	**606 (8%)**	**691 (4%)**	1297 (6%)
Low	7116 (92%)	15119 (96%)	22235 (94%)
Total	7722 (33%)	15810 (67%)	23532 (100%)

2. Subpopulation of size 2,198
Context={State: NY, Race: Black, Gender: Male}
p-value=7.72e-05 ; NMI=[0.0040, 0.0975]

Price	Income <$50K	Income >=$50K	Total
High	**52 (4%)**	**8 (1%)**	60 (3%)
Low	1201 (96%)	937 (99%)	2138 (97%)
Total	1253 (57%)	945 (43%)	2198 (100%)

Reports for Fairness bugs

- Example: simulation of location based pricing scheme
- Test for **disparate impact on low-income populations**
  - Low effect over whole US population
  - High effects in specific sub-populations

Report of associations of O=Price on S_i=Income:
Assoc. metric: norm. mutual information (NMI).

Global Population of size 494,436
p-value=3.34e-10 ; NMI=[0.0001, 0.0005]

Price	Income <$50K	Income >=$50K	Total
High	**15301 (6%)**	**13867 (6%)**	29168 (6%)
Low	234167(94%)	231101(94%)	465268 (94%)
Total	249468(50%)	244968(50%)	494436(100%)

1. Subpopulation of size 23,532
Context={State: CA, Race: White}
p-value=2.31e-24 ; NMI=[0.0051, 0.0203]

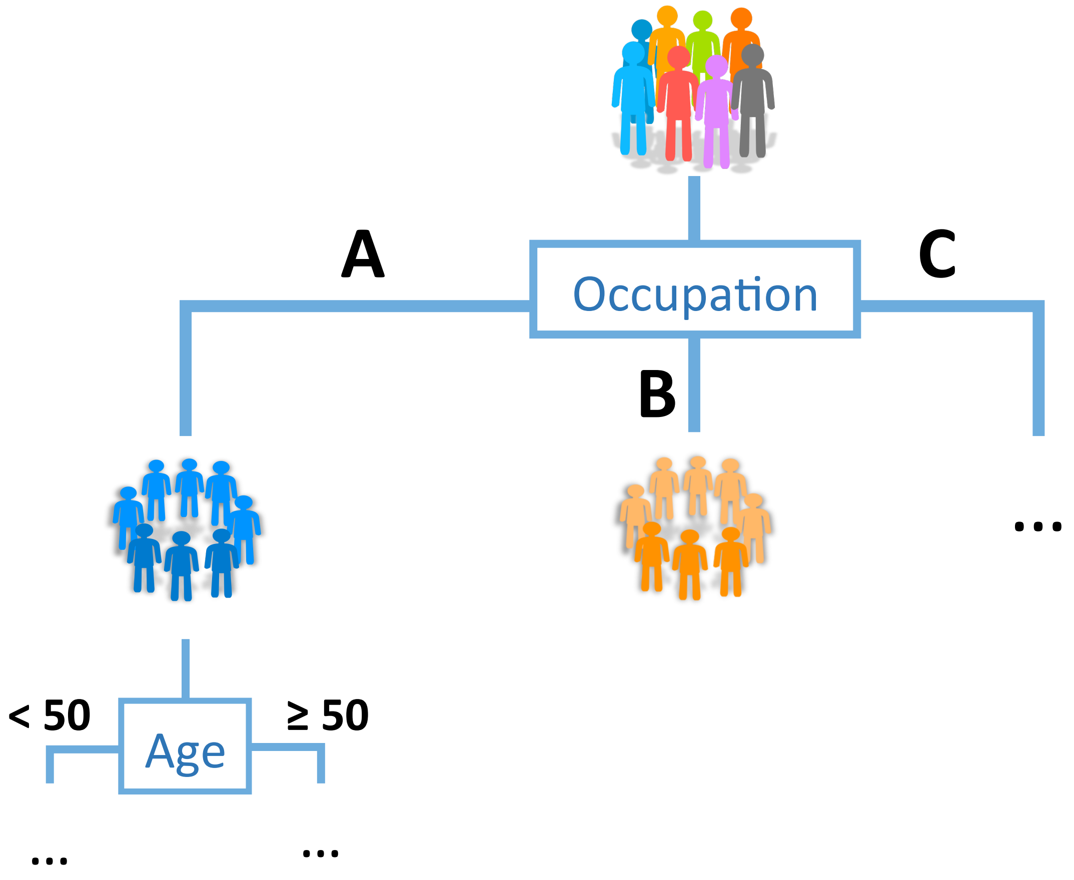
Price	Income <$50K	Income >=$50K	Total
High	**606 (8%)**	**691 (4%)**	1297 (6%)
Low	7116(92%)	15119(96%)	22235 (94%)
Total	7722(33%)	15810(67%)	23532(100%)

2. Subpopulation of size 2,198
Context={State: NY, Race: Black, Gender: Male}
p-value=7.72e-05 ; NMI=[0.0040, 0.0975]

Price	Income <$50K	Income >=$50K	Total
High	**52 (4%)**	**8 (1%)**	60 (3%)
Low	1201(96%)	937(99%)	2138 (97%)
Total	1253(57%)	945(43%)	2198(100%)

Association-Guided Decision Trees

Goal: find most strongly affected user sub-populations



Split into sub-populations with Increasingly strong associations between **protected variables** and **application outputs**

Association-Guided Decision Trees

- Efficient discovery of contexts with high associations
- Outperforms previous approaches based on *frequent itemset mining*
- Easily interpretable contexts by default
- Association-metric agnostic

Metric	Use Case
Binary ratio/difference	Binary variables
Mutual Information	Categorical variables
Pearson Correlation	Scalar variables
Regression	High dimensional outputs
Plugin your own!	**???**

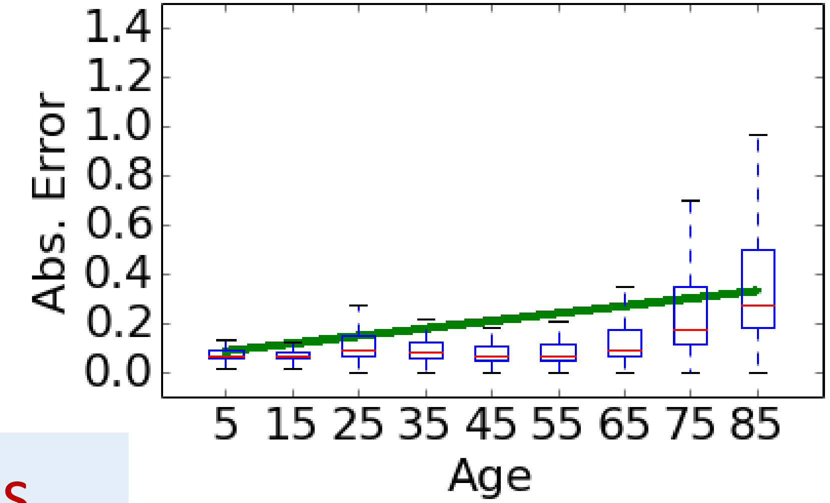
- Greedy strategy (some bugs could be missed)

Example: healthcare application

Predictor of whether patient will visit hospital again in next year
(from winner of 2012 Heritage Health Prize Competition)

FairTest findings: strong association between age and prediction error rate

age, gender,
emergencies, … → Hospital re-admission predictor → Will patient be re-admitted?



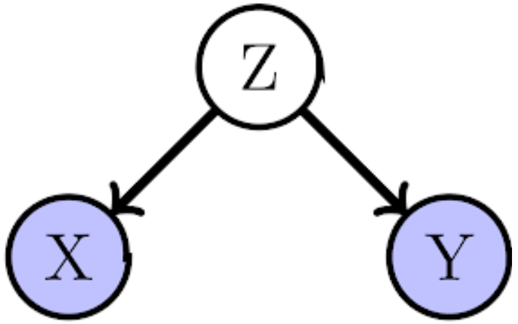
Association may translate to quantifiable harms
(e.g., if model is used to adjust insurance premiums)

Debugging with FairTest

Are there **confounding factors**?

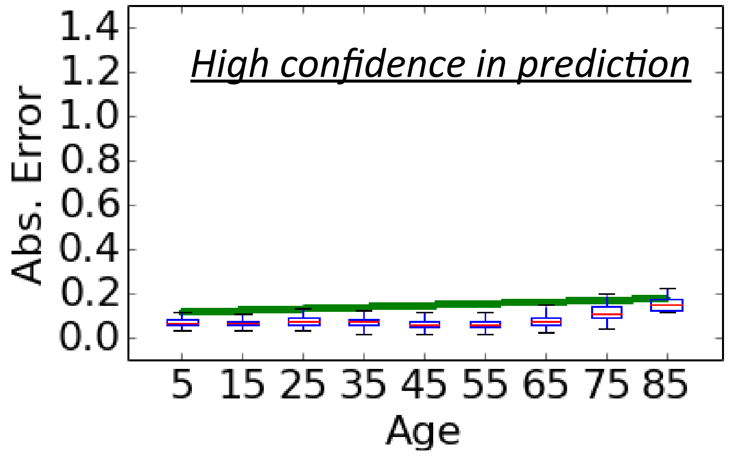
Do associations disappear after conditioning?

$\Rightarrow$ Adaptive Data Analysis!



Example: the healthcare application (again)

- Estimate prediction confidence (target variance)
- Does this **explain** the predictor's behavior?
- Yes, partially



FairTest helps developers understand & evaluate potential association bugs.

Other applications studied using FairTest

- Image tagger based on ImageNet data
  - ⇒ Large output space (~1000 labels)
  - ⇒ FairTest automatically switches to regression metrics
  - ⇒ Tagger has *higher error rate* for pictures of black people

- Simple movie recommender system
  - ⇒ Men are assigned movies with *lower ratings* than women
  - ⇒ Use personal preferences as **explanatory factor**
  - ⇒ FairTest finds no significant bias anymore

Closing remarks

The *Unwarranted Associations* Framework

- Captures a broader set of algorithmic biases than in prior work
- Principled approach for statistically valid investigations

FairTest

- The first end-to-end system for evaluating algorithmic fairness

Developers need better statistical training and tools to make better statistical decisions and applications.

http://arxiv.org/abs/1510.02377

Thanks!

Example: Berkeley graduate admissions

Admission into UC Berkeley graduate programs
(Bickel, Hammel, and O'Connell, 1975)

Bickel *et al*'s (and also FairTest's) findings: gender bias in admissions at university level, but mostly gone after conditioning on department

FairTest helps developers understand & evaluate potential association bugs.