Proxy Attribute Discovery in Machine Learning Datasets via Inductive Logic Programming

Rafael Gonçalves, Filipe Gouveia, Inês Lynce, José Fragoso Santos
INESC-ID & Instituto Superior Técnico, Universidade de Lisboa, Portugal

TACAS '25, Hamilton, Canada

Bias in Machine Learning is widespread

Bias in Machine Learning is widespread

October 11, 2018

Amazon Scraps Secret AI Recruiting Engine that Showed Biases Against Women

AI Research scientists at Amazon uncovered biases against women on their recruiting machine learning engine

Bias in Machine Learning is widespread

October 11, 2018

Amazon Scraps Secret AI Recruiting Engine that Showed Biases Against Women

AI Research scientists at Amazon uncovered biases against women on their recruiting machine learning engine

Exclusive: Age, disability, marital status and nationality influence decisions to investigate claims, prompting fears of ‘hurt first, fix later’ approach

Bias in Machine Learning is widespread

October 11, 2018

Amazon Scraps Secret AI Recruiting Engine that Showed Biases Against Women

AI Research scientists at Amazon uncovered biases against women on their recruiting machine learning engine

Exclusive: Age, disability, marital status and nationality influence decisions to investigate claims, prompting fears of 'hurt first, fix later' approach

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

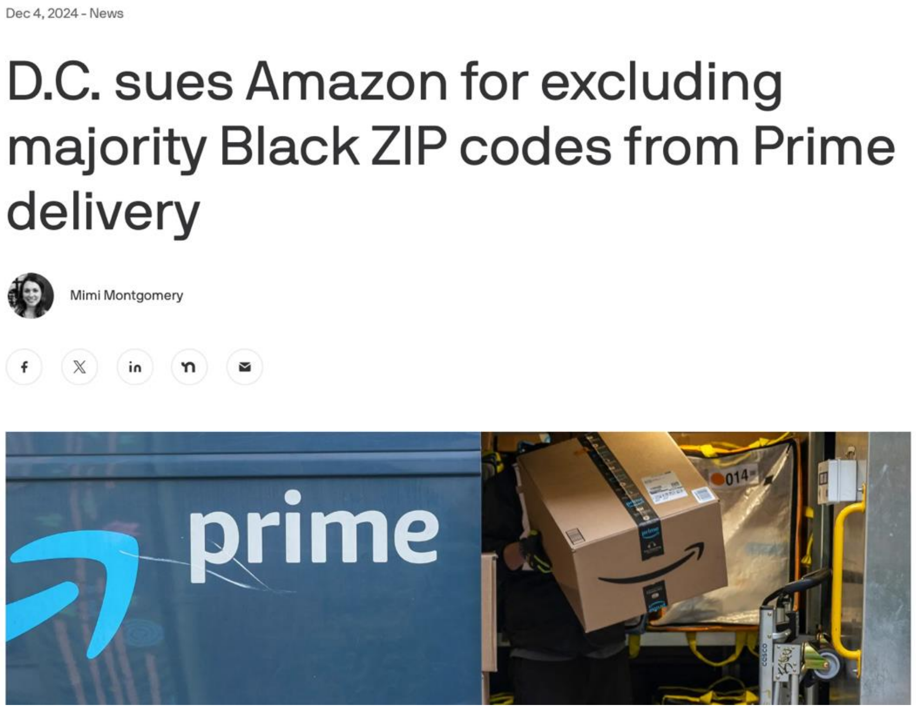
by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

Bias can persist despite mitigation efforts

Common mitigation: avoid making decisions based on protected information

Issue: indirect discrimination


Dec 4, 2024 - News

D.C. sues Amazon for excluding majority Black ZIP codes from Prime delivery

Mimi Montgomery

Bias can persist despite mitigation efforts

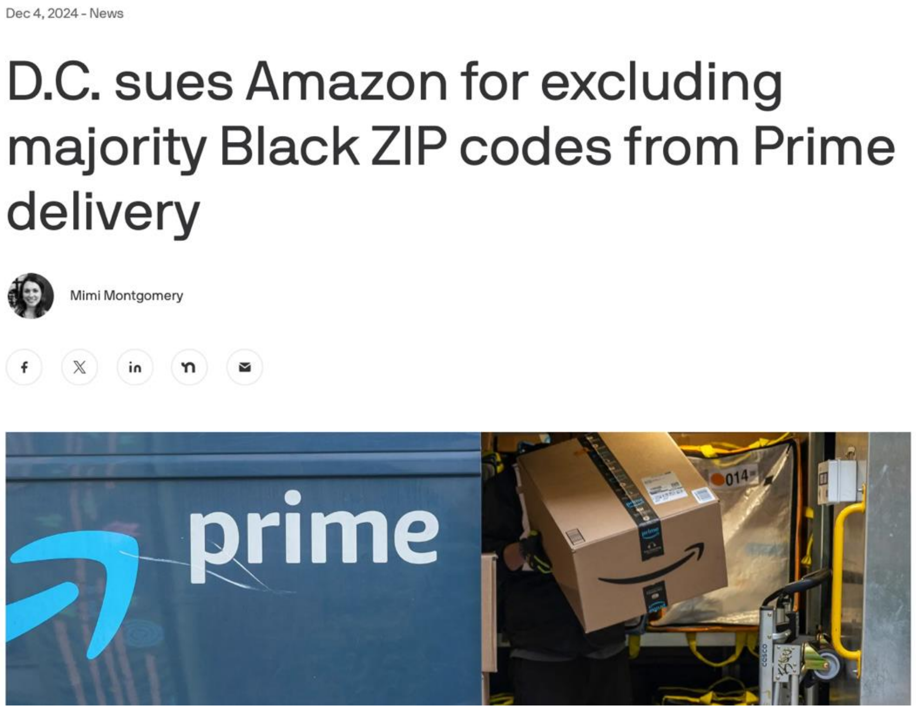
Common mitigation: avoid making decisions based on protected information

Issue: indirect discrimination

- Non-protected data might disclose information about protected data

Proxy attributes

ZIP code discloses race



Proxy attribute discovery: causal graphs

Model datasets as causal graphs and detect relations [Kilbertus17,Kusner17]

Kilbertus et al. Avoiding discrimination through causal reasoning. NIPS '17.
Kusner et al. Counterfactual fairness. NIPS '17.

Proxy attribute discovery: causal graphs

Model datasets as causal graphs and detect relations [Kilbertus17,Kusner17]

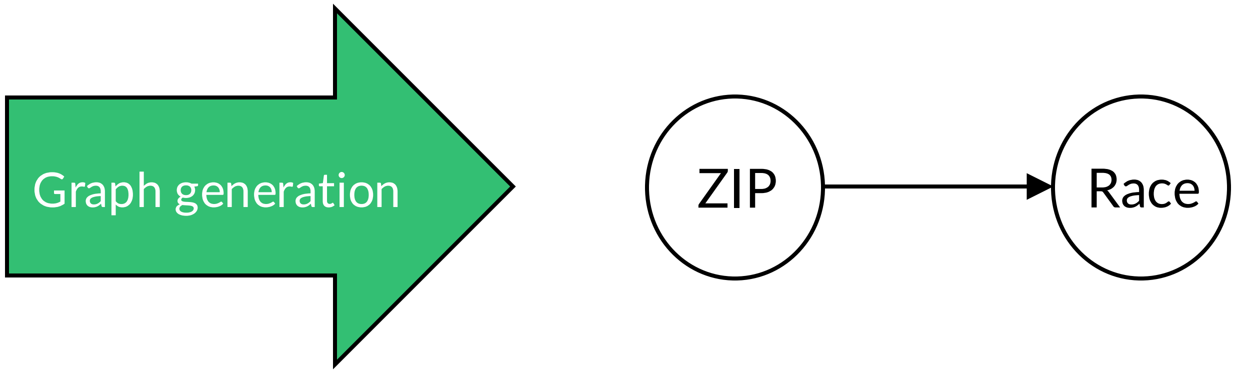
ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Kilbertus et al. Avoiding discrimination through causal reasoning. NIPS ‘17.
Kusner et al. Counterfactual fairness. NIPS ‘17.

Proxy attribute discovery: causal graphs

Model datasets as causal graphs and detect relations [Kilbertus17,Kusner17]

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

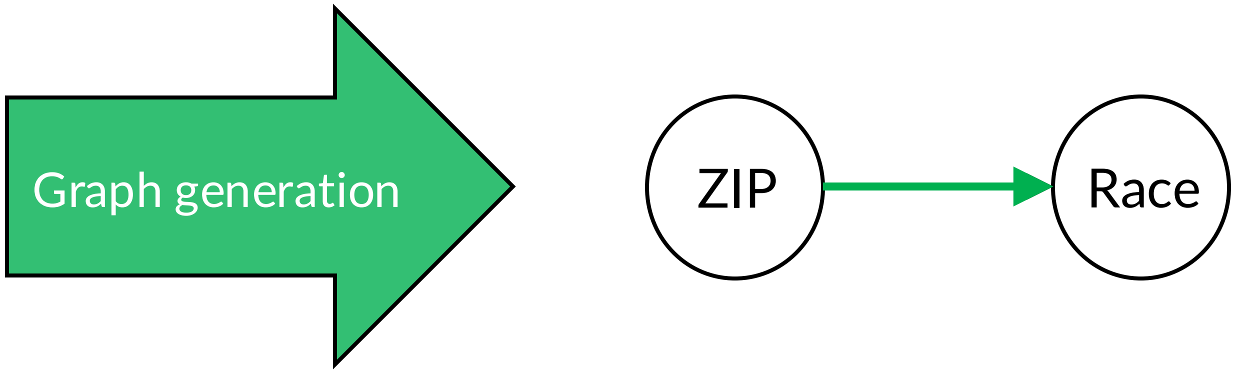


Kilbertus et al. Avoiding discrimination through causal reasoning. NIPS ‘17.
Kusner et al. Counterfactual fairness. NIPS ‘17.

Proxy attribute discovery: causal graphs

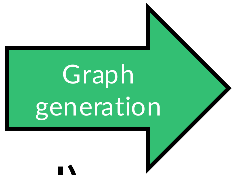
Model datasets as causal graphs and detect relations [Kilbertus17,Kusner17]

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White



Kilbertus et al. Avoiding discrimination through causal reasoning. NIPS ‘17.
Kusner et al. Counterfactual fairness. NIPS ‘17.

Causal graphs: challenges

C1: How do we define 

?

- User-provided (standard)
- Learned from the dataset [LeQuy22]

C2: No support for arithmetic relations beyond equality

- Need to categorize numeric attributes

C3: Lack of expressivity of the output

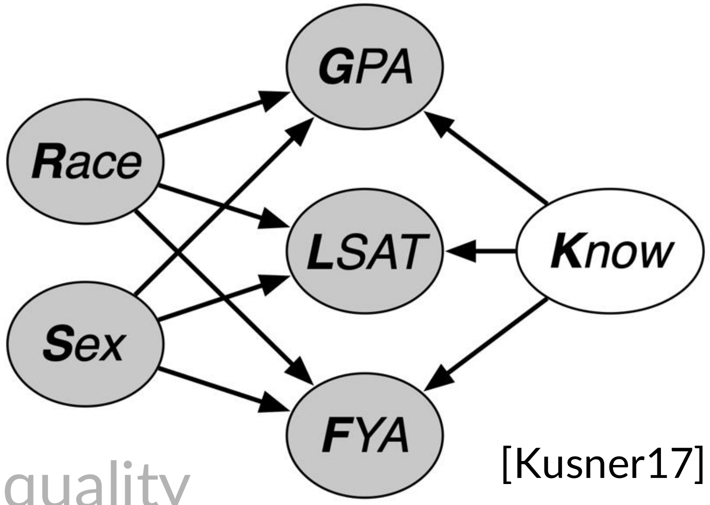
- No insight into underlying proxy relation

Le Quy et al. A survey on datasets for fairness-aware machine learning. WIREs Data Mining and Knowledge Discovery 12(3).

Causal graphs: challenges

C1: How do we define Graph generation ?

- User-provided (standard)
- Learned from the dataset [LeQuy22]



[Kusner17]

C2: No support for arithmetic relations beyond equality

- Need to categorize numeric attributes

C3: Lack of expressivity of the output

- No insight into underlying proxy relation

Le Quy et al. A survey on datasets for fairness-aware machine learning. WIREs Data Mining and Knowledge Discovery 12(3).
Kusner et al. Counterfactual fairness. NIPS '17.

Causal graphs: challenges

C1: How do we define Graph generation ?

- User-provided (standard)
- Learned from the dataset [LeQuy22]

C2: No support for arithmetic relations beyond equality

- Need to categorize numeric attributes

C3: Lack of expressivity of the output

- No insight into underlying proxy relation

Le Quy et al. A survey on datasets for fairness-aware machine learning. WIREs Data Mining and Knowledge Discovery 12(3).

Causal graphs: challenges

C1: How do we define Graph generation ?

- User-provided (standard)
- Learned from the dataset [LeQuy22]

C2: No support for arithmetic relations beyond equality

- Need to categorize numeric attributes

C3: Lack of expressivity of the output

- No insight into underlying proxy relation

Le Quy et al. A survey on datasets for fairness-aware machine learning. WIREs Data Mining and Knowledge Discovery 12(3).

PADTAI: Proxy attribute discovery via Inductive Logic Programming (ILP)

Goal: Learn if/how the protected attribute can be computed from the non-protected attributes

How? Model proxy attribute discovery as ILP problem and infer rules

PADTAI (Proxy Attribute Discovery for Trustworthy AI)

ID	NP_1	NP_2	P
1	…	…	…
2	…	…	…
3	…	…	…
4	…	…	…
5	…	…	…

Encoding

ILP problem

ILP solver

P(X,Y) :- NP_1(X, Z), …
P(X,Y) :- NP_1(X, W), …
P(X,Y) :- NP_2(X, V), …

PADTAI: Proxy attribute discovery via Inductive Logic Programming (ILP)

Goal: Learn if/how the protected attribute can be computed from the non-protected attributes

How? Model proxy attribute discovery as ILP problem and infer rules

PADTAI (Proxy Attribute Discovery for Trustworthy AI)

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

PADTAI: Proxy attribute discovery via Inductive Logic Programming (ILP)

Goal: Learn if/how the protected attribute can be computed from the non-protected attributes

How? Model proxy attribute discovery as ILP problem and infer rules

PADTAI (Proxy Attribute Discovery for Trustworthy AI)

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding → ILP problem

PADTAI: Proxy attribute discovery via Inductive Logic Programming (ILP)

Goal: Learn if/how the protected attribute can be computed from the non-protected attributes

How? Model proxy attribute discovery as ILP problem and infer rules

PADTAI (Proxy Attribute Discovery for Trustworthy AI)

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding → ILP problem → ILP solver →

```
race(X, black) :-
    zipcode(X, 15341)
```

How do we encode proxy attribute discovery as an ILP problem?

Encoding: general idea

Goal: Given dataset, infer rules that compute the protected attribute from the non-protected attributes

Encoding: general idea

Goal: Given dataset, infer rules that compute the protected attribute from the non-protected attributes

What does the ILP solver expect as input?

- **Decision points:** values that the solver can use to make a decision
- **Column relations:** what we already know
- **Examples:** what we're trying to predict

Encoding: general idea

Goal: Given dataset, infer rules that compute the protected attribute from the non-protected attributes

What do we need to encode?

- **Decision points:** distinct values appearing in the dataset
- **Column relations:** what the non-protected attributes of each row are
- **Examples:** what the protected attribute of each row is

Encoding by example

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Input: ZIP Code/Race dataset

Goal: Infer rules that compute **Race** from **ZIP Code**

Encoding by example

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Input: ZIP Code/Race dataset

Goal: Infer rules that compute **Race** from **ZIP Code**

Encoding by example: decision points

Goal: Encode distinct values appearing in the dataset

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: decision points

Goal: Encode distinct values appearing in the dataset

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: decision points

Goal: Encode distinct values appearing in the dataset

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding

```
pwhite(white).
pblack(black).
p15341(15341).
p15823(15823).
p15782(15782).
```

Encoding by example: column relations

Goal: Encode non-protected attributes of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: column relations

Goal: Encode non-protected attributes of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: column relations

Goal: Encode non-protected attributes of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding

```
pzipcode(1,15341).
pzipcode(2,15823).
pzipcode(3,15341).
pzipcode(4,15341).
pzipcode(5,15782).
```

Encoding by example: examples

Goal: Encode protected attribute of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: examples

Goal: Encode protected attribute of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding by example: examples

Goal: Encode protected attribute of each row

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White

Encoding

```
pos(prace(1,black)).
pos(prace(2,white)).
pos(prace(3,black)).
pos(prace(4,black)).
pos(prace(5,white)).
```

Full encoding and inferred rule

Goal: Infer rules that compute **Race** from **ZIP Code**

```
pwhite(white).
pblack(black).
p15341(15341).
p15823(15823).
p15782(15782).
```

```
pzipcode(1,15341).
pzipcode(2,15823).
pzipcode(3,15341).
pzipcode(4,15341).
pzipcode(5,15782).
```

```
pos(prace(1,black)).
pos(prace(2,white)).
pos(prace(3,black)).
pos(prace(4,black)).
pos(prace(5,white)).
```

Full encoding and inferred rule

Goal: Infer rules that compute **Race** from **ZIP Code**

```
pwhite(white).
pblack(black).
p15341(15341).
p15823(15823).
p15782(15782).
```

```
pzipcode(1,15341).
pzipcode(2,15823).
pzipcode(3,15341).
pzipcode(4,15341).
pzipcode(5,15782).
```

```
pos(prace(1,black)).
pos(prace(2,white)).
pos(prace(3,black)).
pos(prace(4,black)).
pos(prace(5,white)).
```

ILP solver

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

X is black if their ZIP Code is 15341

Full encoding and inferred rule

Goal: Infer rules that compute **Race** from **ZIP Code**

```
pwhite(white).
pblack(black).
p15341(15341).
p15823(15823).
p15782(15782).
```

```
pzipcode(1,15341).
pzipcode(2,15823).
pzipcode(3,15341).
pzipcode(4,15341).
pzipcode(5,15782).
```

```
pos(prace(1,black)).
pos(prace(2,white)).
pos(prace(3,black)).
pos(prace(4,black)).
pos(prace(5,white)).
```

ILP solver

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

X is black if their ZIP Code is 15341

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

ID	ZIP Code	Race
1	15341	Black
2	15341	White
3	15341	Black

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

ID	ZIP Code	Race
1	15341	Black
2	15341	White
3	15341	Black

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

ID	ZIP Code	Race
1	15341	Black
2	15341	White
3	15341	Black

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

ID	ZIP Code	Race
1	15341	Black
2	15341	White
3	15341	Black

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- Need to validate inferred rules against entire dataset

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Encoding: challenges and preprocessing

C1: How do we handle conflicting rows?

- Plurality voting

C2: How do we scale to large datasets?

- Sampling
- **Need to validate inferred rules against entire dataset**

C3: Support for arithmetic relations beyond equality

- Encoding is parametric on set of arithmetic relations
- Support for less-than operator by default
- User can extend with additional operations

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

R = 3/4

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute occurs

Higher is better (ideally: >10/15%)

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

P = 3/4

$$P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows where the protected attribute is predicted to occur

Higher is better (ideally: >90%)

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

```
race(X, Y) :-
    zipcode(X, Z),
    p15341(Z),
    pblack(Y)
```

$$C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows

Thresholds

Filter rules to ensure they are general, accurate and statistically significant

ID	ZIP Code	Race
1	15341	Black
2	15823	White
3	15341	Black
4	15341	Black
5	15782	White
6	15245	Black
7	15341	White

race(X, Y) :-
zipcode(X, Z),
p15341(Z),
pblack(Y)

C = 3/7

$$C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Fraction of rows where we can correctly predict the protected attribute among all rows

Higher is better (ideally: >5/10%)

EQ: Can PADTAI find proxy attributes in real-world datasets?

Experimental setup

Ran PADTAI with **Popper** [Cropper21] ILP solver on 10 datasets

Dataset	**Task**	**Protected**	**# Proxies**
Adult	*income* > US$50 000?	*race, sex, age*	2
KDD	*income* > US$50 000?	*sex, race*	2
German Credit	*credit risk?*	*age, foreign-worker*	1
Bank Marketing	*makes deposit?*	*age, marital*	1
Credit Card	*default risk?*	*education, marriage, sex*	1
COMPAS	*recidivism risk?*	*race, sex*	2
Ricci	*promoted?*	*race*	1
Students	*final year grade?*	*sex, age*	1
OULAD	*final result?*	*gender*	1
Lawschool	*passes bar?*	*gender, race*	2

Cropper and Morel. Learning programs by learning from failures. Machine Learning 110(4).

Experimental setup

Ran PADTAI with **Popper** [Cropper21] ILP solver on 10 datasets

Dataset	Task	Protected	# Proxies
Adult	*income* > US$50 000?	*race, sex, age*	2
KDD	*income* > US$50 000?	*sex, race*	2
German Credit	*credit risk?*	*age, foreign-worker*	1
Bank Marketing	*makes deposit?*	*age, marital*	1
Credit Card	*default risk?*	*education, marriage, sex*	1
COMPAS	*recidivism risk?*	*race, sex*	2
Ricci	*promoted?*	*race*	1
Students	*final year grade?*	*sex, age*	1
OULAD	*final result?*	*gender*	1
Lawschool	*passes bar?*	*gender, race*	2

Cropper and Morel. Learning programs by learning from failures. Machine Learning 110(4).

Experimental setup

Ran PADTAI with **Popper** [Cropper21] ILP solver on 10 datasets

Dataset	Task	Protected	# Proxies
Adult	*income* $>$ US\$50 000?	*race, sex, age*	2
KDD	*income* $>$ US\$50 000?	*sex, race*	2
German Credit	*credit risk?*	*age, foreign-worker*	1
Bank Marketing	*makes deposit?*	*age, marital*	1
Credit Card	*default risk?*	*education, marriage, sex*	1
COMPAS	*recidivism risk?*	*race, sex*	2
Ricci	*promoted?*	*race*	1
Students	*final year grade?*	*sex, age*	1
OULAD	*final result?*	*gender*	1
Lawschool	*passes bar?*	*gender, race*	2

Cropper and Morel. Learning programs by learning from failures. Machine Learning 110(4).

Experimental setup

Ran PADTAI with **Popper** [Cropper21] ILP solver on 10 datasets

Dataset	Task	Protected	# Proxies
Adult	*income* $>$ US$50 000?	*race, sex, age*	2
KDD	*income* $>$ US$50 000?	*sex, race*	2
German Credit	*credit risk?*	*age, foreign-worker*	1
Bank Marketing	*makes deposit?*	*age, marital*	1
Credit Card	*default risk?*	*education, marriage, sex*	1
COMPAS	*recidivism risk?*	*race, sex*	2
Ricci	*promoted?*	*race*	1
Students	*final year grade?*	*sex, age*	1
OULAD	*final result?*	*gender*	1
Lawschool	*passes bar?*	*gender, race*	2

Cropper and Morel. Learning programs by learning from failures. Machine Learning 110(4).

Experimental setup

Ran PADTAI with **Popper** [Cropper21] ILP solver on 10 datasets

Dataset	Task	Protected	# Proxies
Adult	*income* > US$50 000?	*race, sex, age*	2
KDD	*income* > US$50 000?	*sex, race*	2
German Credit	*credit risk?*	*age, foreign-worker*	1
Bank Marketing	*makes deposit?*	*age, marital*	1
Credit Card	*default risk?*	*education, marriage, sex*	1
COMPAS	*recidivism risk?*	*race, sex*	2
Ricci	*promoted?*	*race*	1
Students	*final year grade?*	*sex, age*	1
OULAD	*final result?*	*gender*	1
Lawschool	*passes bar?*	*gender, race*	2

Cropper and Morel. Learning programs by learning from failures. Machine Learning 110(4).

Can PADTAI find proxy attributes?

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	**Recall / Precision / Coverage** (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)				
KDD (2)				
German Credit (1)				
Bank Marketing (1)				
Credit Card (1)				
COMPAS (2)				
Ricci (1)				
Students (1)				
OULAD (1)				
Lawschool (2)				
Total (14)				

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)				
KDD (2)				
German Credit (1)				
Bank Marketing (1)				
Credit Card (1)				
COMPAS (2)				
Ricci (1)				
Students (1)				
OULAD (1)				
Lawschool (2)				
Total (14)				

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)				
KDD (2)				
German Credit (1)				
Bank Marketing (1)				
Credit Card (1)				
COMPAS (2)				
Ricci (1)				
Students (1)				
OULAD (1)				
Lawschool (2)				
Total (14)				

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to 83 proxy relations in 8 datasets with varying thresholds, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Dataset	**Recall / Precision / Coverage** (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to **83** proxy relations in **8** datasets with varying thresholds, corresponding to **49** potential proxy attributes, including 11 previously identified in the literature and **38 new ones**

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to **83** proxy relations in **8** datasets with varying thresholds, corresponding to **49** potential proxy attributes, including 11 previously identified in the literature and **38 new ones**

Can PADTAI find proxy attributes?

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

PADTAI detects up to 83 proxy relations in 8 datasets with **varying thresholds**, corresponding to 49 potential proxy attributes, including 11 previously identified in the literature and 38 new ones

Can PADTAI find proxy attributes?

Support for **numeric attributes**:

race(X, white) :- combine(X, Y), lt(79, Y). [Ricci]
X is white if their combine score is greater than 79

- Missed by prior work because it incorrectly categorized *combine* into categories {<70, ≥70}

Relations with **multiple proxy attributes**:

race(X, white) :- native_country(X, usa), education(X, bsc). [Adult]
X is white if they are from the US and have a bachelor's degree

- Combination of proxies native_country and education was previously unknown

Can PADTAI find proxy attributes?

Support for **numeric attributes**:

race(X, white) :- combine(X, Y), lt(79, Y).   [Ricci]
X is white if their combine score is greater than 79

- Missed by prior work because it incorrectly categorized *combine* into categories {<70, ≥70}

Relations with **multiple proxy attributes**:

race(X, white) :- native_country(X, usa), education(X, bsc).   [Adult]
X is white if they are from the US and have a bachelor's degree

- Combination of proxies native_country and education was previously unknown

Can PADTAI find proxy attributes?

Support for **numeric attributes**:

race(X, white) :- combine(X, Y), lt(79, Y). [Ricci]
X is white if their combine score is greater than 79

- Missed by prior work because it incorrectly categorized *combine* into categories {<70, ≥70}

Relations with **multiple proxy attributes**:

race(X, white) :- native_country(X, usa), education(X, bsc). [Adult]
X is white if they are from the US and have a bachelor's degree

- Combination of proxies native_country and education was previously unknown

Proxy Attribute Discovery in Machine Learning Datasets via Inductive Logic Programming

- ML datasets often suffer from indirect discrimination via **proxy attributes**
- Proposed **PADTAI**, a new tool for proxy attribute discovery based on **ILP**
- Evaluated on 10 real-world datasets and detected 83 proxy relations corresponding to 49 potential proxy attributes
- **PADTAI** is open-source and available online

Link to PADTAI