Program fairness – a formal methods perspective

Aditya V. Nori
Microsoft Research Cambridge

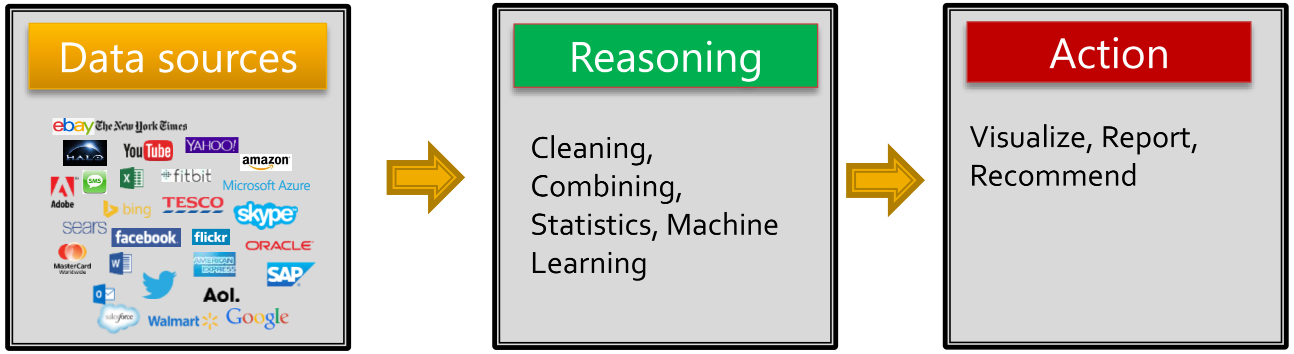
Joint work with Samuel Drews, Aws Albarghouthi, Loris D'Antoni (University of Wisconsin-Madison)

Data is everywhere!

- Data analysis is big part of today’s software
- Increasingly developers creating and using machine learning models
- Increasingly developers working with data that is incomplete, inaccurate, approximate

New programming language challenges

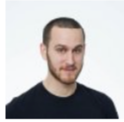
- Bringing data into programs
  - Numerous new sources
  - Conversion



- Reasoning with data
  - *What does correctness mean?*

1949 1960 1970 1980 1990 2000

Amazon just showed us that 'unbiased' algorithms can be inadvertently racist



Rafi Letzter

Apr. 21, 2016, 4:50 PM 1,808 1

Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

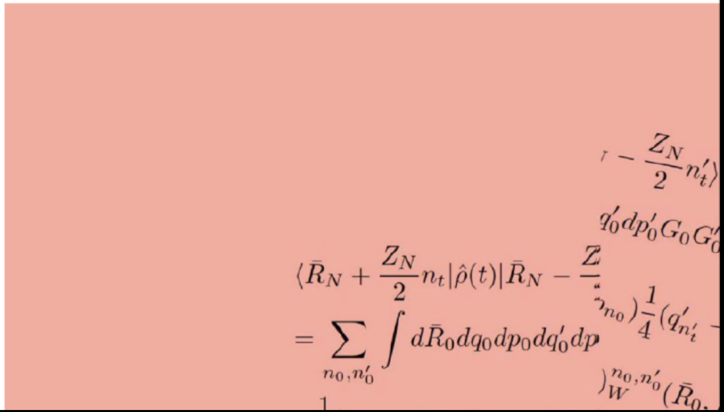
Machine Bias

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

Internet Of Things

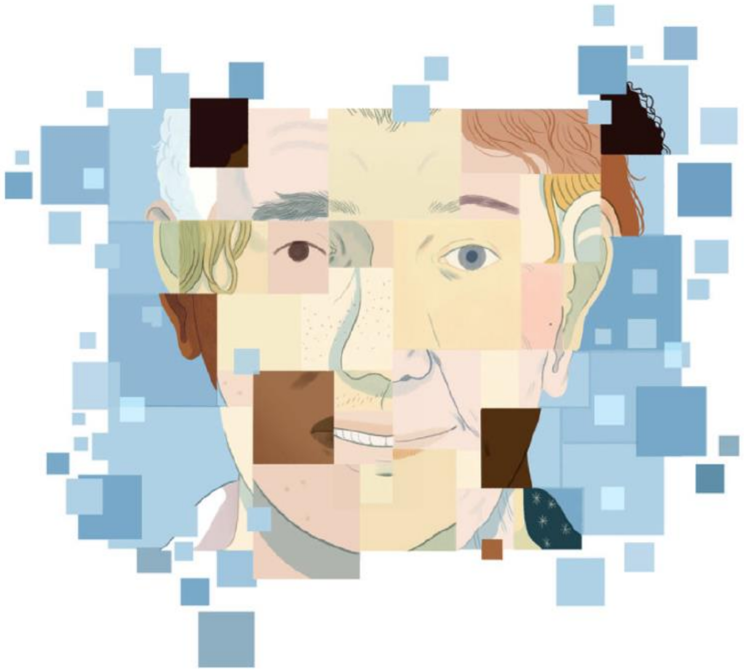
Who do you blame when an algorithm gets you fired?



SundayReview | OPINION

Artificial Intelligence's White Guy Problem

By KATE CRAWFORD JUNE 25, 2016



Bianca Bagnarelli

TheUpshot

HIDDEN BIAS

When Algorithms Discriminate

Claire Cain Miller @clairecm JULY 9, 2015

How do we prove that a program does not discriminate?

How do we prove that a program does not discriminate?

- Theoreticians
  - How do we formalise fairness?
- Machine learning researchers
  - How do we learn fair models?
- Security/privacy researchers
  - How do we detect bias in black-box algorithms
- Legal scholars
  - How do we regulate algorithmic decision making?

This is important!

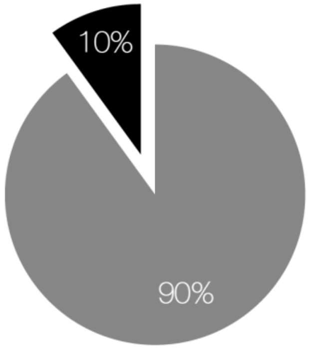
- EU GDPR (2018)
  - "data subject's explicit consent"
  - "right to explanation"
- White House report (2014)
  - "Powerful algorithms … raise the potential of **encoding discrimination in automated decisions**."
- White House report (recently …)
  - "Federal agencies that use AI-based systems to make or provide decision support for consequential decisions about individuals should take extra care to **ensure the efficacy and fairness** of those systems, based on **evidence-based verification and validation**."

Our work

1) Fairness as a program property

2) Automatic proofs of (un)fairness

Algorithmic fairness
[Dwork et al. '12, Zemel et al. '13, Feldman et al. '15]



$$\{v = (v_1, \ldots, v_s, \ldots)\}$$

$$h \leftarrow \mathcal{D}(v)$$

$$\left\{ \frac{\Pr[h \mid v_s]}{\Pr[h \mid \neg v_s]} > 1 - \epsilon \right\}$$

Decision-making program

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\left\{ \frac{\Pr[\texttt{hire} \mid \texttt{ethnicity > 10}]}{\Pr[\texttt{hire} \mid \texttt{ethnicity <= 10}]} > 1 - \epsilon \right\}$$

Decision-making program

population model

$$\{v \sim \mathcal{M}\}$$

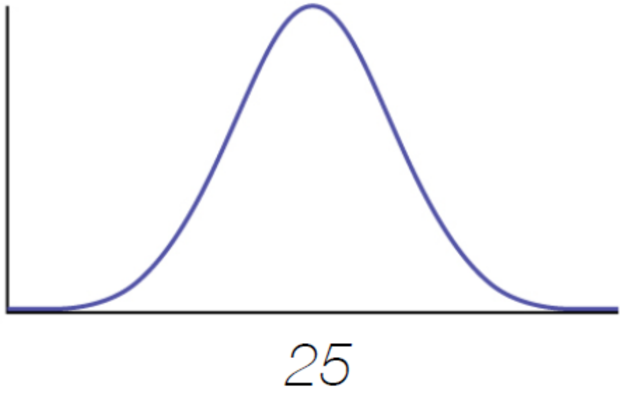
Hoare triple ☺

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\left\{ \frac{\Pr[\texttt{hire} \mid \texttt{ethnicity > 10}]}{\Pr[\texttt{hire} \mid \texttt{ethnicity <= 10}]} > 1 - \epsilon \right\}$$

population model → decision-making program

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```




```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\{v \sim \mathcal{M}\}$$

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\left\{ \frac{\Pr[\texttt{hire} \mid \texttt{ethnicity > 10}]}{\Pr[\texttt{hire} \mid \texttt{ethnicity <= 10}]} > 1 - \epsilon \right\}$$

$$\{v \sim \mathcal{M}\}$$

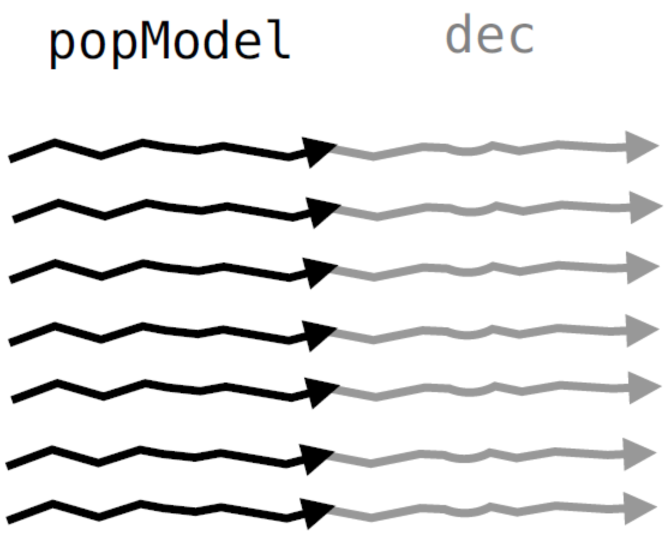
```
def dec(colRank, yExp, ethnicity)
 expRank ← yExp - colRank
 if (colRank <= 5)
  hire ← true
 elif (expRank > -5)
  hire ← true
 else
  hire ← false
 return hire
```

by definition of
conditional probability

$$\left\{ \frac{\Pr[\mathtt{hire} \wedge \mathtt{min}] \cdot \Pr[\neg\mathtt{min}]}{\Pr[\mathtt{hire} \wedge \neg\mathtt{min}] \cdot \Pr[\mathtt{min}]} > 1 - \epsilon \right\}$$

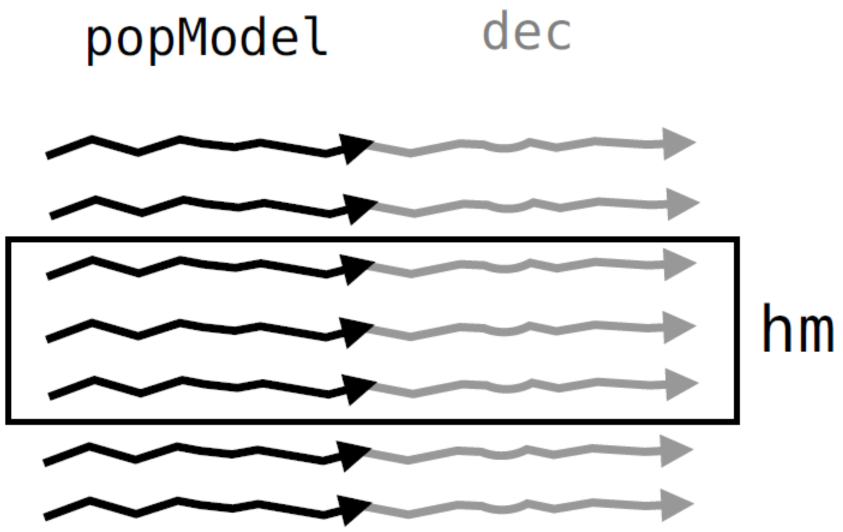
Let's focus on $Pr[hire \wedge min]$

- Π: set of all possible execution paths in dec(popModel())
- $p(\pi)$: probability that $\pi \in \Pi$



$Pr[hire \wedge min]$

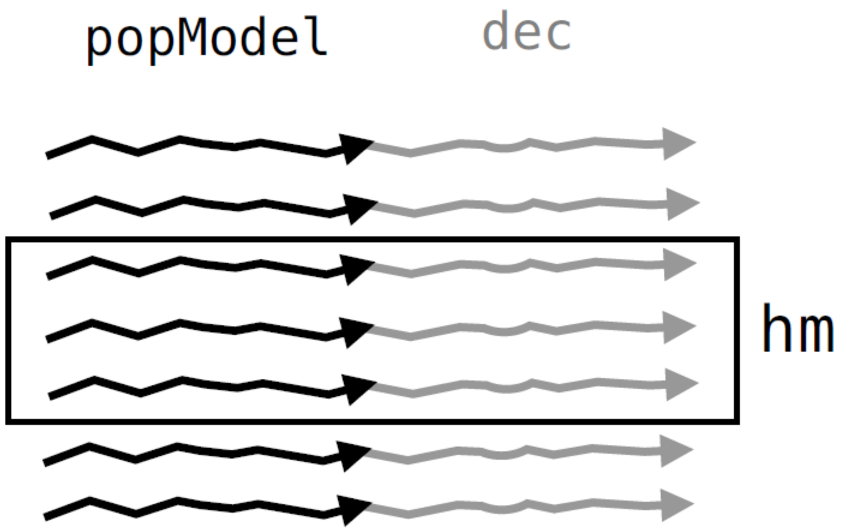
- Π: set of all possible execution paths in dec(popModel())
- $p(\pi)$: probability that $\pi \in \Pi$



$$\boxed{\Pr[\mathtt{hire} \wedge \mathtt{min}] = \sum_{\pi \in \Pi_{\mathsf{hm}}} p(\pi)}$$

$\Pr[\text{hire} \wedge \text{min}]$

- Π: set of all possible execution paths in **dec(popModel())**
- $p(\pi)$: probability that $\pi \in \Pi$



$$\boxed{\Pr[\mathtt{hire} \wedge \mathtt{min}] = \int_{\Pi_{\mathsf{hm}}} p(\pi)\, d\pi}$$

What does this mean?

$$\int_{\Pi_{hm}} p(\pi) d\pi$$

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\int_{\Pi_{hm}} p(\pi) d\pi$$

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```

→ Each path is **uniquely represented** by 3 real values

Idea

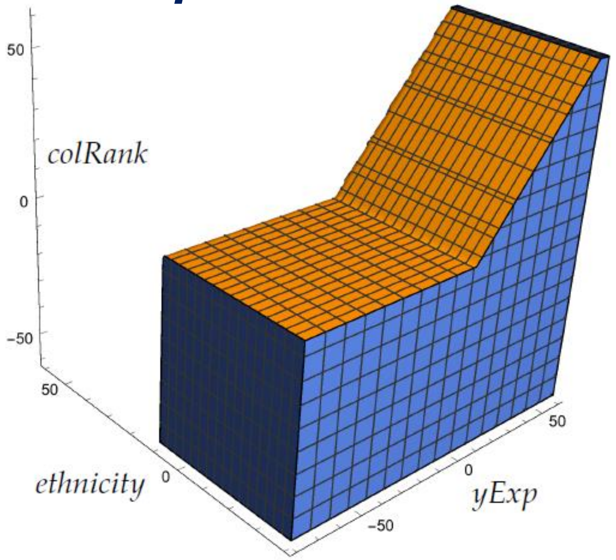
represent paths Π_{hm} as a region $\varphi \subseteq \mathbb{R}^3$

and compute: $$\int_{\varphi} p_e(e) p_c(c) p_y(y)\ de\ dp\ dy$$

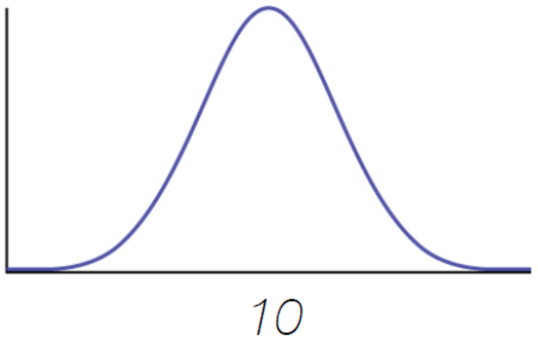
Weighted volume: $\int_{\varphi} p_e(e)p_c(c)p_y(y)\, de\, dp\, dy$

- Volume: $\int_{\varphi} 1\, de\, dp\, dy$

φ



$p_y(y)$



How do we define φ?

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\varphi_{\text{pop}} \equiv \mathit{ethnicity} > 10 \Rightarrow \mathit{colRank}_1 = \mathit{colRank} + 5$$
$$\wedge\, \mathit{ethnicity} \leqslant 10 \Rightarrow \mathit{colRank}_1 = \mathit{colRank}$$

How do we define φ?

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\varphi_{\text{pop}} \equiv ethnicity > 10 \Rightarrow colRank_1 = colRank + 5$$
$$\wedge\, ethnicity \leqslant 10 \Rightarrow colRank_1 = colRank$$

How do we define φ?

```
def popModel()
  ethnicity ~ gauss(0,10)
  colRank ~ gauss(25,10)
  yExp ~ gauss(10,5)
  if (ethnicity > 10)
    colRank ← colRank + 5
  return colRank, yExp, ethnicity
```

```
def dec(colRank, yExp, ethnicity)
  expRank ← yExp - colRank
  if (colRank <= 5)
    hire ← true
  elif (expRank > -5)
    hire ← true
  else
    hire ← false
  return hire
```

$$\varphi_{\mathrm{pop}} \equiv \boxed{ethnicity > 10 \Rightarrow colRank_1 = colRank + 5} \wedge ethnicity \leqslant 10 \Rightarrow colRank_1 = colRank$$

$$\varphi_{\mathrm{dec}} \equiv expRank = yExp^{i} - colRank^{i} \wedge hire \iff (colRank^{i} \leqslant 5 \vee expRank > -5)$$

$$\varphi \equiv \exists V_d .\ \varphi_{\texttt{pop}} \wedge \varphi_{\texttt{dec}} \wedge \texttt{hire} \wedge \texttt{min}$$

all non-probabilistic variables
(deterministic variables)

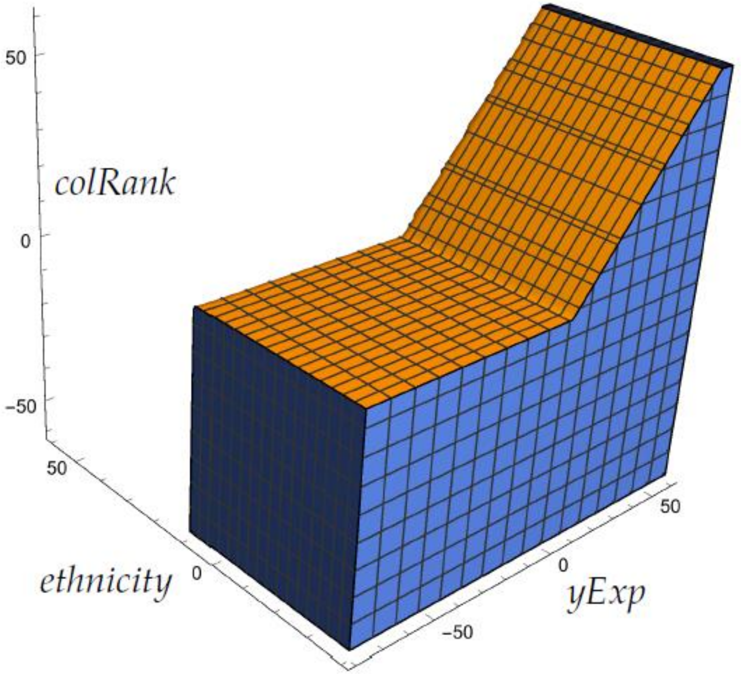
To compute $Pr[hire \wedge min]$

- Represent all executions as an SMT formula φ
- Compute the weighted volume of φ

$$\int_{\varphi} p_e(e) p_c(c) p_y(y) \, de \, dp \, dy$$

Volume of a polytope is #P-hard [Dyer and Frieze, 1988]

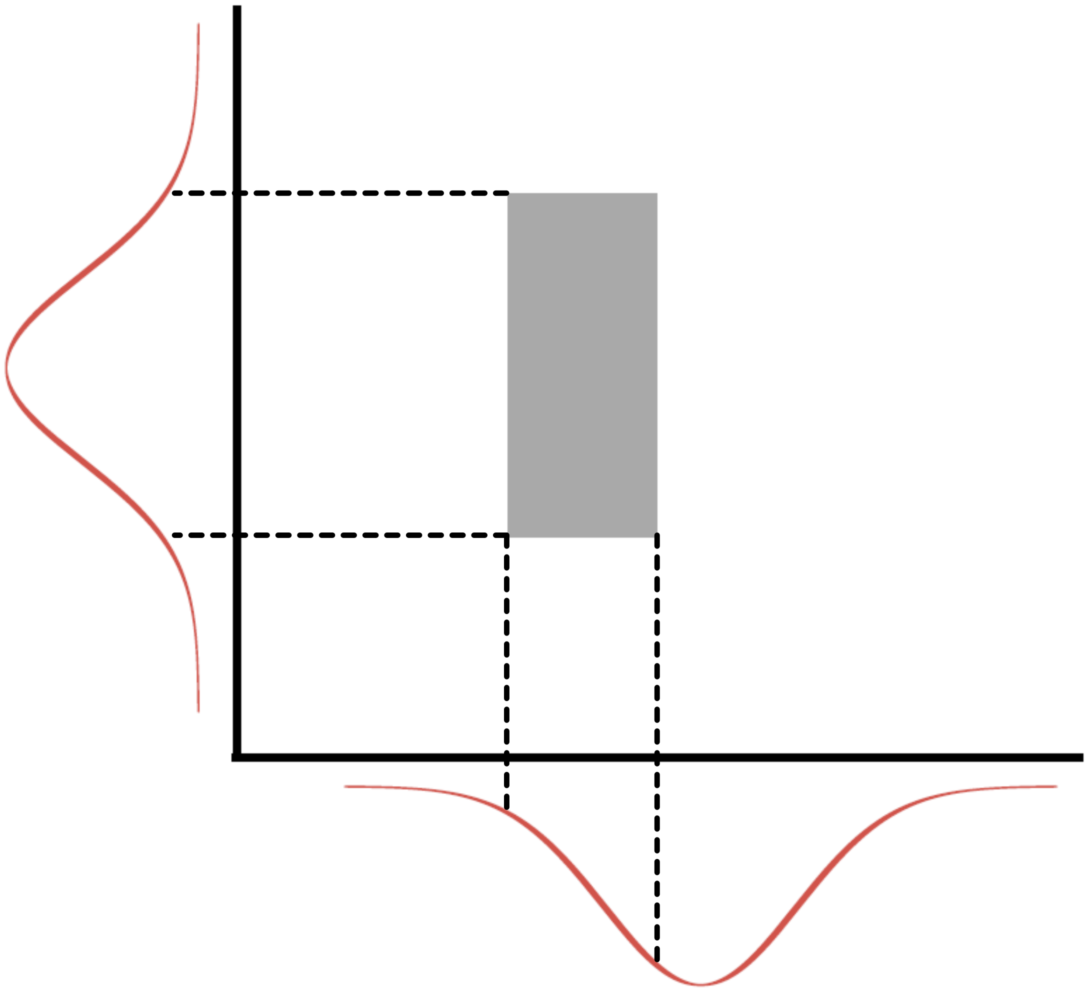
Weighted volume computation



$$\int_{\varphi} p_e(e)\, p_c(c)\, p_y(y)\; de\; dp\; dy$$

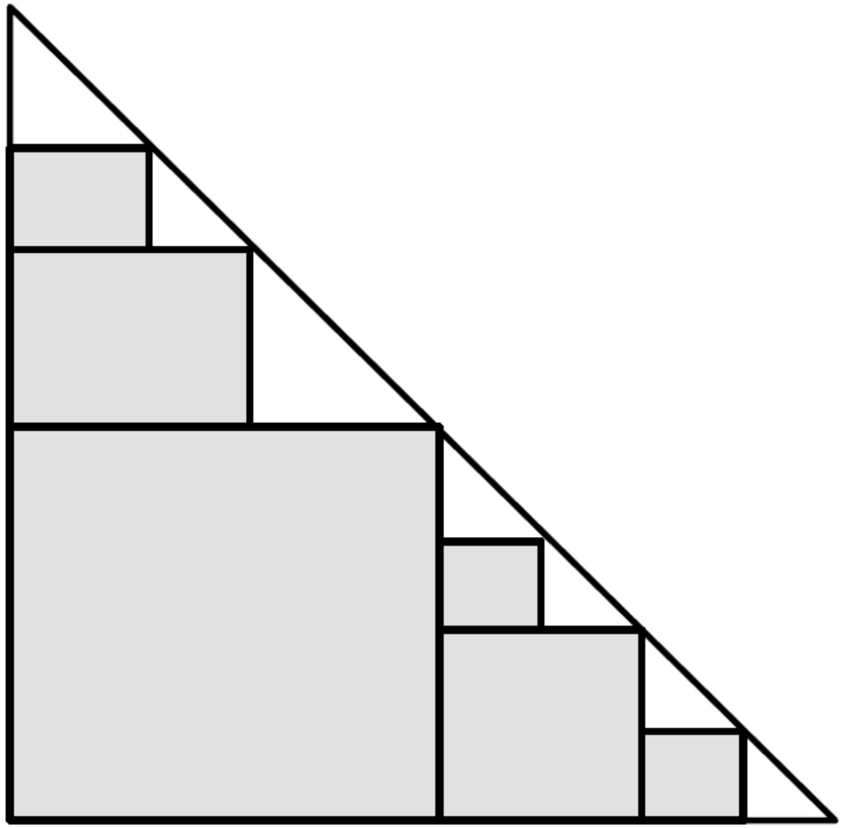
Observation

- Rectangles are easy!



What about this triangle?

- There are **infinitely many rectangles**



General idea

- Hyperrectangular decomposition
  - consider all hyperrectangles in φ

- Hyperrectangular sampling
  - Iteratively sample $H \Rightarrow \varphi$

Hyperrectangular decompostion

let $\mathcal{H}$ be the set of all hyperrectangles in φ

construct formula $\square_{\varphi}$ s.t.

- there is a one-to-one map between $\mathcal{H}$ and models of $\square_{\varphi}$

Hyperrectangular decompostion

all variables

lower/upper bounds

$$\square_{\varphi} \equiv \forall \mathcal{X}_{\varphi}.\left(\bigwedge_{x \in \mathcal{X}_{\varphi}} l_x \leqslant x \leqslant u_x\right) \Rightarrow \varphi$$

$$\wedge \left(\bigwedge_{x \in \mathcal{X}_{\varphi}} l_x \leqslant u_x\right)$$

Hyperrectangular sampling

$vol \leftarrow 0$

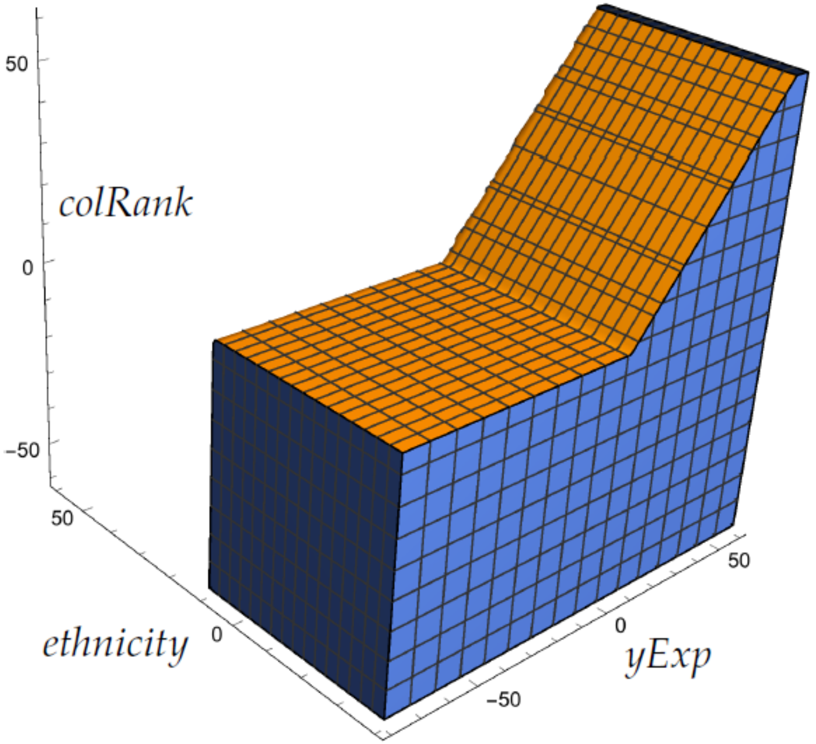
get a model $m \models \square_{\varphi}$

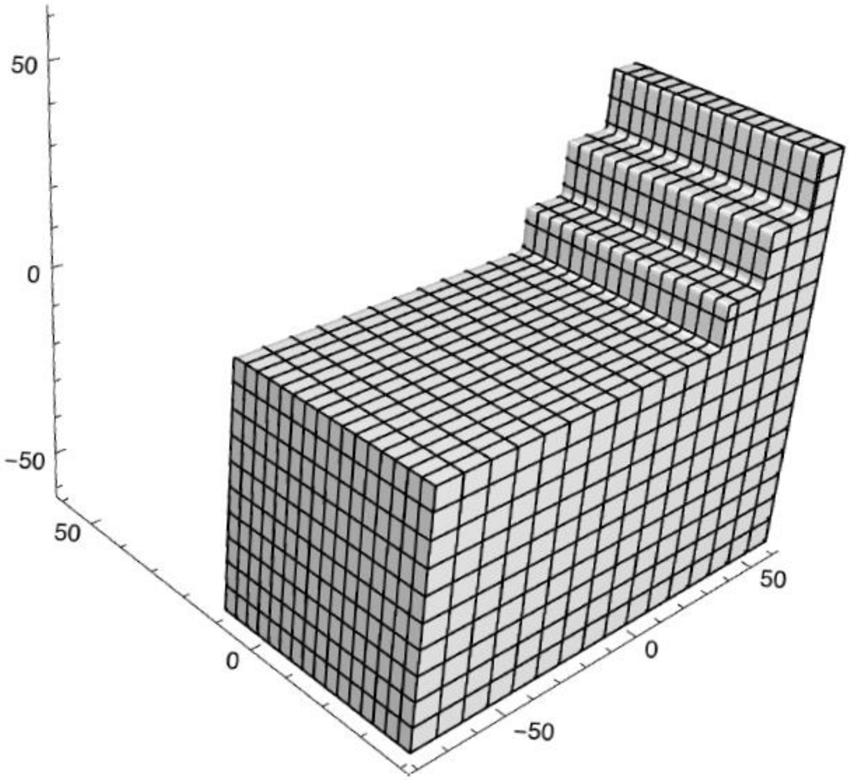
$vol \leftarrow vol + \textsc{vol}(H^m)$

block all rectangles overlapping with H^m and repeat

we know how to compute this

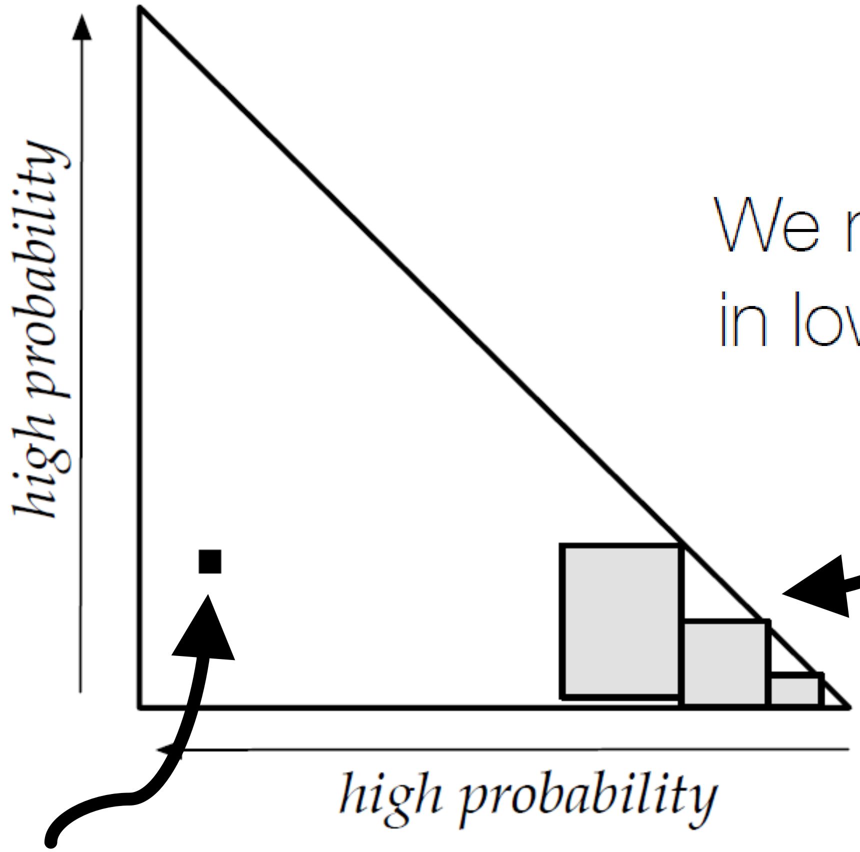
Hyperrectangular sampling





what could go wrong?

Sampling challenges



We might keep sampling in low-probability regions

unit rectangles have no volume

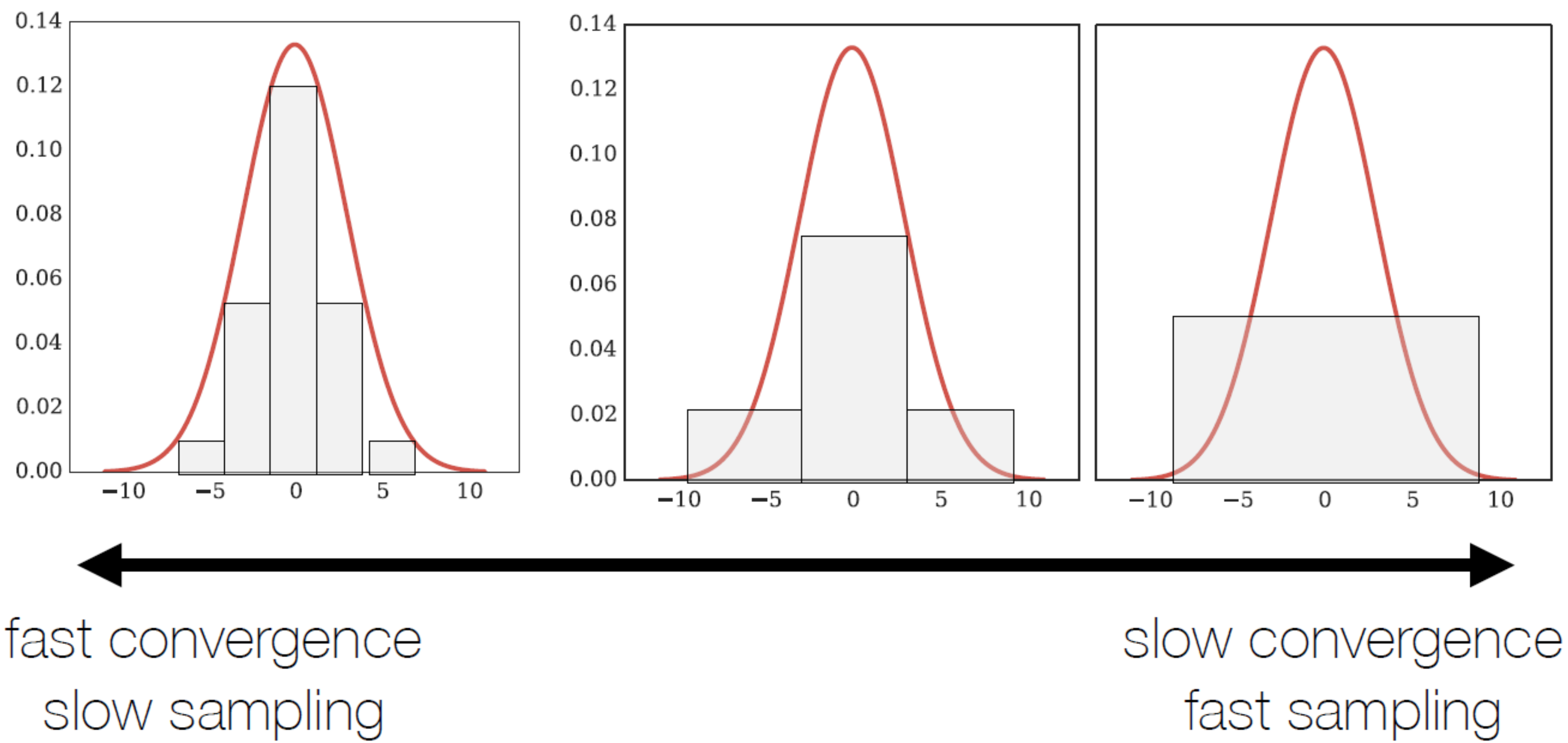
Density-directed sampling

- **Ideal solution:** sample with the following objective

$$\underset{m \models \Box_{\varphi}}{\arg\max} \ \mathrm{VOL}(H^m)$$

Solution that works

- Approximate densities with step functions
  - *area under a step function is a linear formula*



Properties of the algorithm

- Maintains a lower-bound on volume
- Converges to the actual volume in the limit
- Works for real closed fields
- To compute upper-bound, negate formula

$$\textsc{vol}(\varphi) = 1 - \textsc{vol}(\neg\varphi)$$

$$\textsc{vol}(\varphi) \leqslant 1 - \textsc{vol}_{under}(\neg\varphi)$$

Proof of fairness

lower bound on numerator

$$\left\{ \frac{\Pr[\texttt{hire} \mid \texttt{ethnicity > 10}]}{\Pr[\texttt{hire} \mid \texttt{ethnicity <= 10}]} > 1 - \epsilon \right\}$$

upper bound on denominator

Proof of unfairness

upper bound on numerator

$$\left\{ \frac{\Pr[\texttt{hire} \mid \texttt{ethnicity > 10}]}{\Pr[\texttt{hire} \mid \texttt{ethnicity <= 10}]} > 1 - \epsilon \right\}$$

lower bound on denominator

Evaluation

Decision program	Acc	Population Model: Independent			Clusters			Bayes Net		
		Res	*Vol*	*QE*	*Res*	*Vol*	*QE*	*Res*	*Vol*	*QE*
DT_4	0.79	✓	1.3	0.5	✗	10.0	3.7	✗	2.2	0.9
DT_{14}	0.71	✓	4.2	1.4	✓	106.3	19.3	✓	52.3	11.4
DT_{16}	0.79	✓	7.7	2.0	✗	44.5	44.4	✗	8.9	7.6
DT_{44}	0.82	✓	63.5	9.8	0.22 1.63	TO	843.4	0.70 0.88	TO	165.0
SVM_3	0.79	✓	2.6	0.6	✗	20.8	4.8	✗	3.7	1.7
SVM_4	0.79	✓	2.7	0.8	✗	45.7	5.8	✗	6.5	2.7
SVM_5	0.79	✓	8.5	1.3	✗	28.3	8.6	✗	54.3	5.4
SVM_6	0.79	0.02 35.3	TO	2.4	0.04 86.7	TO	10.4	0.09 3.03	TO	12.8
$\mathrm{NN}_{2,1}$	0.65	✓	21.6	0.8	0.70 0.97	TO	3.9	✓	456.1	3.4
$\mathrm{NN}_{2,2}$	0.67	✓	27.8	2.0	0.70 0.98	TO	11.7	✓	236.5	7.2
$\mathrm{NN}_{3,2}$	0.74	0.03 674.7	TO	10.0	0.12 7.71	TO	101.3	0.00 5.24	TO	55.9
$\mathrm{DT}^{\alpha}_{16}$	0.76	✓	5.1	3.0	✓	233.8	88.9	✓	93.6	10.6
$\mathrm{SVM}^{\alpha}_{4}$	0.78	✓	3.0	0.8	✓	103.9	5.6	✓	735.2	3.2

Summary

- Automatic proofs of (un)fairness for decision making programs

- Future directions
  - Scalability – application to real-world programs
  - Explaining unfairness
  - Repairing unfair programs

[FairSquare: Probabilistic Verification for Program Fairness]
Aws Albarghouthi, Loris D'Antoni, Samuel Drews, Aditya Nori, OOPSLA '17