Lecture 16: Fairness Verification

CIS 7000: Trustworthy Machine Learning

Spring 2024

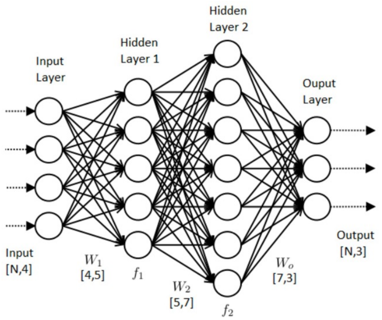
Agenda

- Fairness verification problem
- Symbolic fairness verification
- Statistical fairness verification

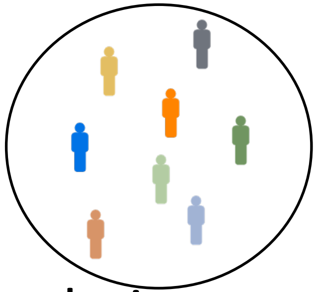
Fairness Verification

- **Goal:** Check if a given model satisfies a given fairness definition
- Ideally, the verification strategy should be flexible, and work on a broad family of fairness definitions
  - Focus on group fairness
- **Note:** Fairness is a **statistical property**!
  - Depends on data distribution $p(x, y)$
  - Therefore, we also need to specify $p(x, y)$, which we call the **population model**

Fairness Verification

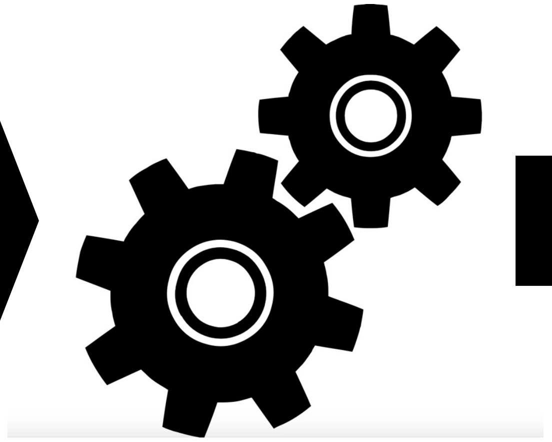


ML model

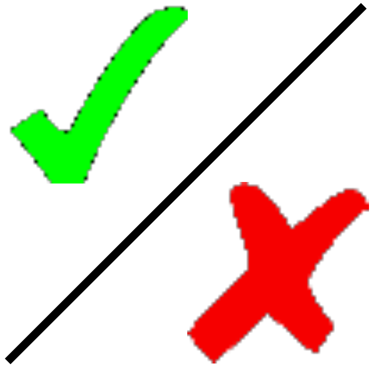


population model

fairness definition



verification algorithm



fair/unfair

Fairness

- **Problem Setup**
    - Distribution $P_{\mathcal{V}}$ over individuals $v = (\tilde{v}, a) \in \mathcal{V}$ (called the **population model**)
    - Sensitive attribute $a \in \{\text{majority}, \text{minority}\}$
    - Binary classifier $f: \mathcal{V} \to \{0,1\}$, where 1 indicates a positive outcome
- **Fairness Properties:** Demographic parity, equality of opportunity, etc.

Demographic Parity

- Majority and minority members get positive outcomes at the same rate
- Let the **acceptance probability** for a be

$$\mu_a^* = \Pr_{v \sim \mathcal{V}}[\, f(v) = 1 \mid A = a \,]$$

- Then, f satisfies demographic parity if $Y_{\text{parity}}^* = 1$, where

$$Y_{\text{parity}}^* = 1\left[\frac{\mu_{\text{minority}}^*}{\mu_{\text{majority}}^*} \geq c\right]$$

  - The constant $c \in [0,1]$ is domain specific
  - **Question:** Does $Y_{\text{parity}}^* = 1$?

Fairness Verification Problem

```
def population_model():
   is_male ~ bernoulli(0.5)
   col_rank ~ normal(25, 10)
   if is_male:
      years_exp ~ normal(15, 5)
   else:
      years_exp ~ normal(10, 5)
   return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
   if col_rank <= 5:
      return true
   elif years_exp > 5:
      return true
   else:
      return false
```

Question: Does OfferJob satisfy demographic parity?

Agenda

- Fairness verification problem
- Symbolic fairness verification
- Statistical fairness verification

Fairness Verification

- **Goal:** Check if $Y^*_{\text{parity}} = 1$, where

$$Y^*_{\text{parity}} = 1\left[\frac{\mu^*_{\text{minority}}}{\mu^*_{\text{majority}}} \geq 1\right]$$

$$\mu^*_a = \Pr_{v \sim \mathcal{V}}[\,f(v) = 1 \mid A = a\,]$$

- **Step 1:** Compute approximation $\hat{\mu}_a \approx \mu^*_a$
- **Step 2:** Compute approximation $\hat{Y}_{\text{parity}} \approx Y^*_{\text{parity}}$

Fairness Verification Strategy

```
def population_model():
   is_male ~ bernoulli(0.5)
   col_rank ~ normal(25, 10)
   if is_male:
      years_exp ~ normal(15, 5)
   else:
      years_exp ~ normal(10, 5)
   return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
   if col_rank <= 5:
      return true
   elif years_exp > 5:
      return true
   else:
      return false
```

Question: Does OfferJob satisfy demographic parity?

Fairness Verification Strategy

```
def population_model():
    is_male ~ bernoulli(0.5)
    col_rank ~ normal(25, 10)
    if is_male:
        years_exp ~ normal(15, 5)
    else:
        years_exp ~ normal(10, 5)
    return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
    if col_rank <= 5:
        return true
    elif years_exp > 5:
        return true
    else:
        return false
```

Question: What is $\Pr\left[\text{OfferJob}\left(\text{PopulationModel}(\,)\right) \mid \text{IsMale} = \text{True}\right]$?

Fairness Verification Strategy

```
def population_model():
    is_male ~ bernoulli(0.5)
    col_rank ~ normal(25, 10)
    if is_male:
        years_exp ~ normal(15, 5)
    else:
        years_exp ~ normal(10, 5)
    return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
    if col_rank <= 5:
        return true
    elif years_exp > 5:
        return true
    else:
        return false
```

Question: What is $\Pr[\text{OfferJob} \mid \text{IsMale} = \text{True}]$?

Fairness Verification Strategy

```
def population_model():
    is_male ~ bernoulli(0.5)
    col_rank ~ normal(25, 10)
    if is_male:
        years_exp ~ normal(15, 5)
    else:
        years_exp ~ normal(10, 5)
    return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
    if col_rank <= 5:
        return true
    elif years_exp > 5:
        return true
    else:
        return false
```

Question: What is $\Pr[\text{OfferJob}]$?

Fairness Verification Strategy

$\Pr[\text{OfferJob}]$

$= \int \text{OfferJob}(a, r, e) \cdot p_{\text{IsMale}}(a) \cdot p_{\text{ColRank}}(r) \cdot p_{\text{YearsExp}}(e) \cdot da \cdot dr \cdot de$

$\text{OfferJob} = (\text{ColRank} \leq 5 \vee \text{YearsExp} > 5)$

$= (\text{ColRank} \leq 5) \vee (\text{ColRank} > 5 \wedge \text{IsMale} \wedge \text{YearsExpLarge} > 5)$

$\vee (\text{ColRank} > 5 \wedge \neg\text{IsMale} \wedge \text{YearsExpSmall} > 5)$

$\Pr[\text{OfferJob}]$

$= \Pr[\text{ColRank} \leq 5] + \Pr[\text{ColRank} > 5 \wedge \text{IsMale} \wedge \text{YearsExpLarge} > 5]$

$+ \Pr[\text{ColRank} > 5 \wedge \neg\text{IsMale} \wedge \text{YearsExpSmall} > 5]$

$= \Pr[\text{ColRank} \leq 5] + \Pr[\text{ColRank} \leq 5] \cdot \Pr[\text{IsMale}] \cdot \Pr[\text{YearsExpLarge} > 5]$

$+ \Pr[\text{ColRank} \leq 5] \cdot \Pr[\neg\text{IsMale}] \cdot \Pr[\text{YearsExpSmall} > 5]$

$= N(5; 25{,}10) + \left(1 - N(5; 25{,}10)\right) \cdot 0.5 \cdot \left(1 - N(5; 15{,}5)\right)$

$+ \left(1 - N(5; 25{,}10)\right) \cdot 0.5 \cdot \left(1 - N(5; 10{,}5)\right)$

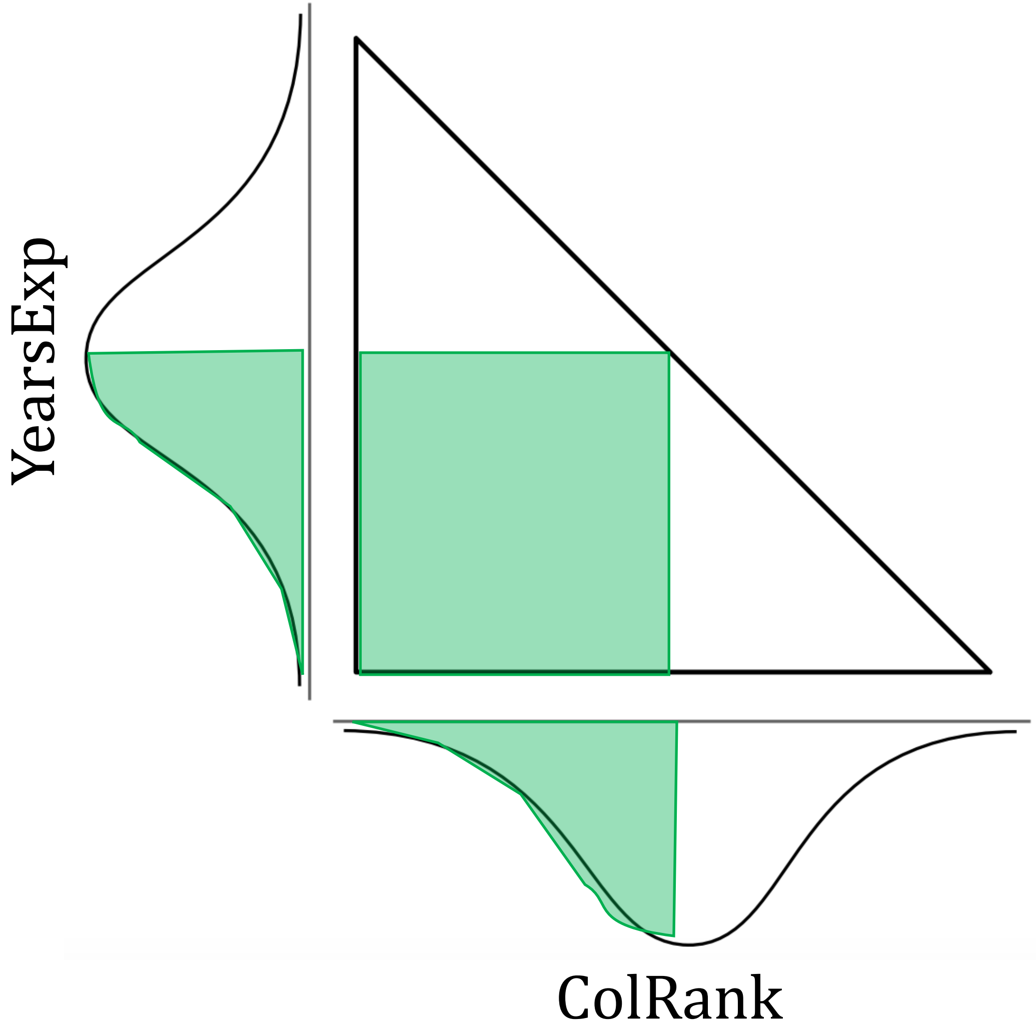
```
def population_model():
    is_male ~ bernoulli(0.5)
    col_rank ~ normal(25, 10)
    if is_male:
        years_exp ~ normal(15, 5)
    else:
        years_exp ~ normal(10, 5)
    return col_rank, years_exp
```

```
def offer_job(col_rank, years_exp)
    if col_rank <= 5:
        return true
    elif years_exp > 5:
        return true
    else:
        return false
```

Fairness Verification Strategy

- **Alternative example**
  - $\text{OfferJob} = (\text{ColRank} + \text{YearsExp}) \leq 10$
  - Assume $\text{ColRank} \sim N(25,10)$ and $\text{YearsExp} \sim N(15,5)$
  - **Goal:** Compute $\Pr[\text{OfferJob}]$

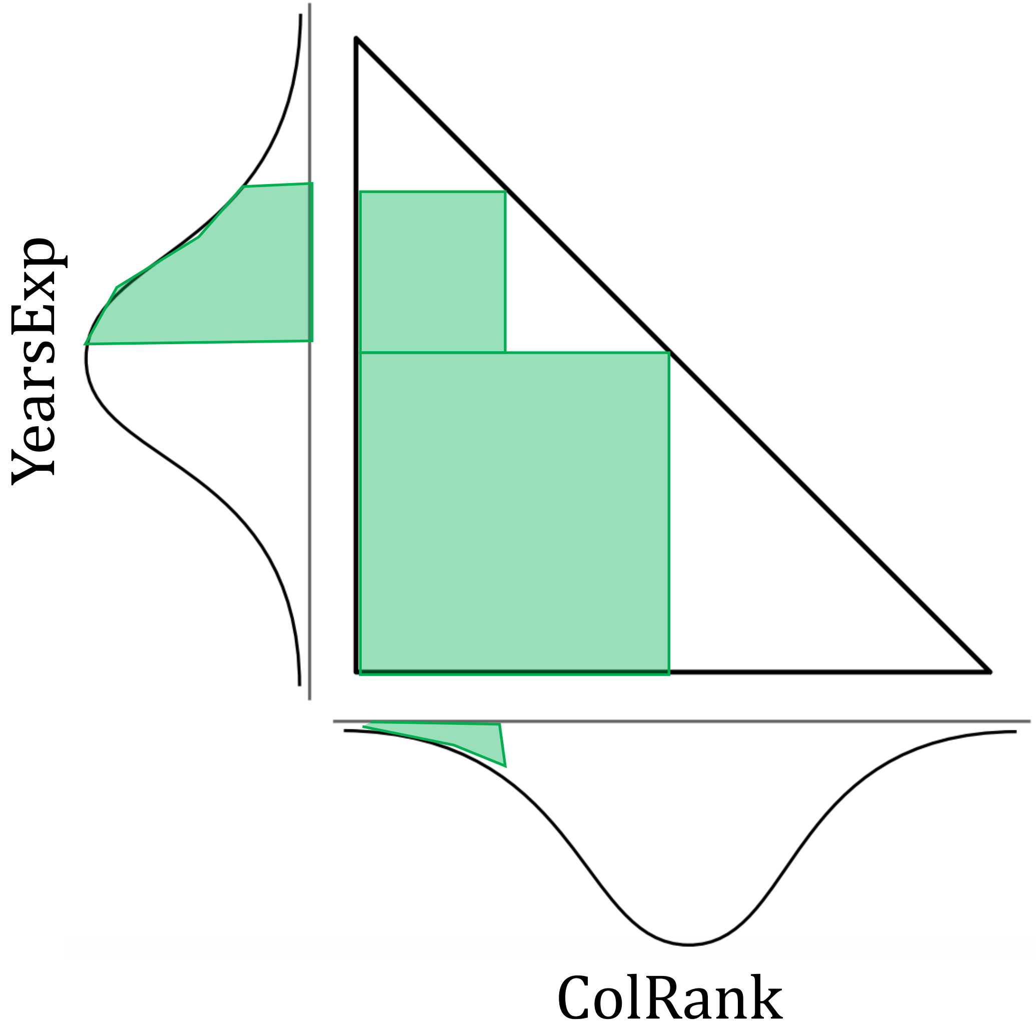
- **Idea:** Break OfferJob into hyperrectangles
  - $R_1 = 0 \leq \text{ColRank} \leq 5 \wedge 0 \leq \text{YearsExp} \leq 5$
  - $\Pr[R_1] = \big(N(5;25,10) - N(0;25,10)\big) \cdot \big(N(5;15,5) - N(0;15,5)\big)$



Fairness Verification Strategy

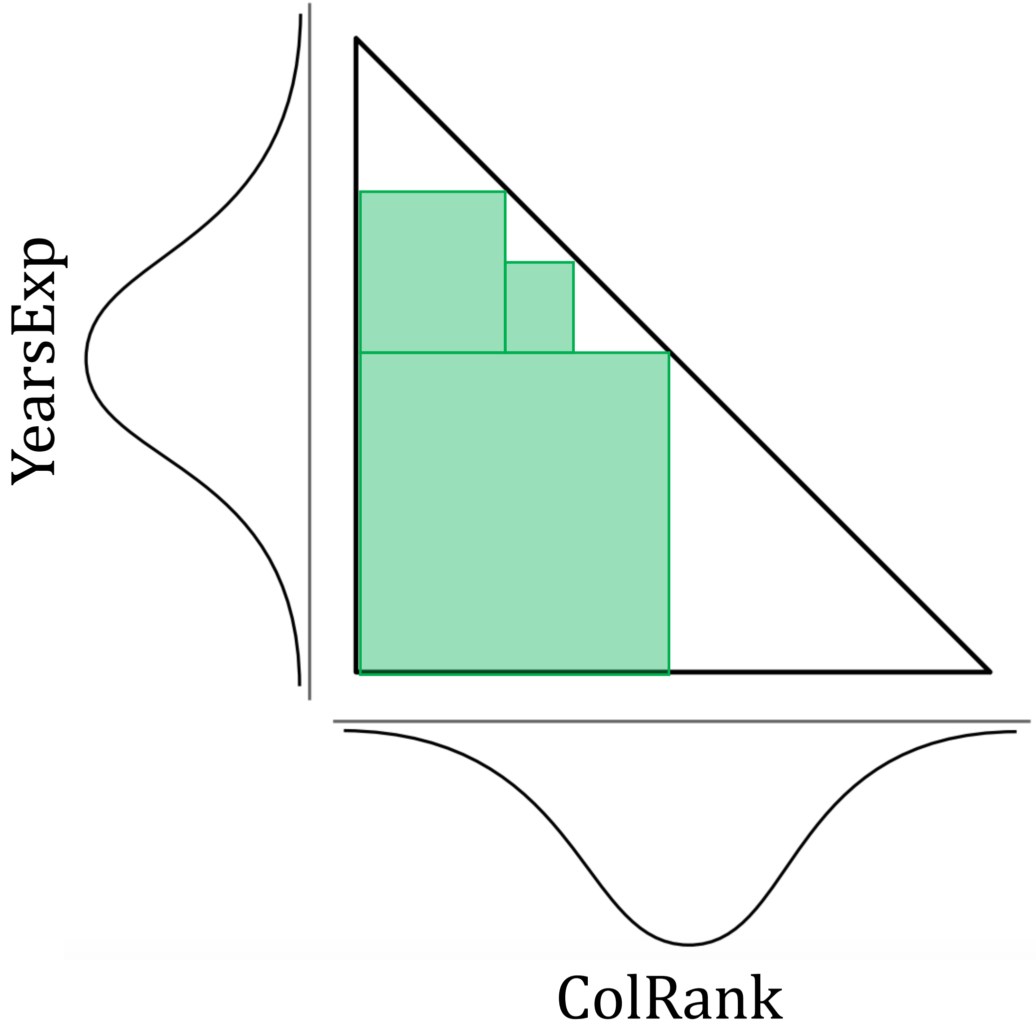
- **Alternative example**
  - $\text{OfferJob} = (\text{ColRank} + \text{YearsExp}) \leq 10$
  - Assume $\text{ColRank} \sim N(25,10)$ and $\text{YearsExp} \sim N(15,5)$
  - **Goal:** Compute $\Pr[\text{OfferJob}]$

- **Idea:** Break OfferJob into hyperrectangles
  - $R_2 = 0 \leq \text{ColRank} \leq 2 \land 5 \leq \text{YearsExp} \leq 8$
  - $\Pr[R_2] = (N(2;25,10) - N(0;25,10)) \cdot (N(8;15,5) - N(5;15,5))$



Fairness Verification Strategy

- **Alternative example**
  - $\text{OfferJob} = (\text{ColRank} + \text{YearsExp}) \leq 10$
  - Assume $\text{ColRank} \sim N(25,10)$ and $\text{YearsExp} \sim N(15,5)$
  - **Goal:** Compute $\Pr[\text{OfferJob}]$



- **Idea:** Break OfferJob into hyperrectangles
  - $\Pr[\text{OfferJob}] = \Pr[R_1] + \Pr[R_2] + \cdots$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

  for $i \in \{1,\ldots,k\}$

    compute rectangle $R_{i,t}$ for ϕ_i

    compute estimate $\hat{\mu}_a \approx \mu_a^*$ using $R_{i,t}$

    compute estimate $\hat{Y} \approx Y_{\text{parity}}^*$ using $\hat{\mu}_a$

    if converged: **return** $\hat{Y}$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

    for $i \in \{1,\ldots,k\}$

        compute rectangle $R_{i,t}$ for ϕ_i

        compute estimate $\hat{\mu}_a \approx \mu_a^*$ using $R_{i,t}$

        compute estimate $\hat{Y} \approx Y_{\mathrm{parity}}^*$ using $\hat{\mu}_a$

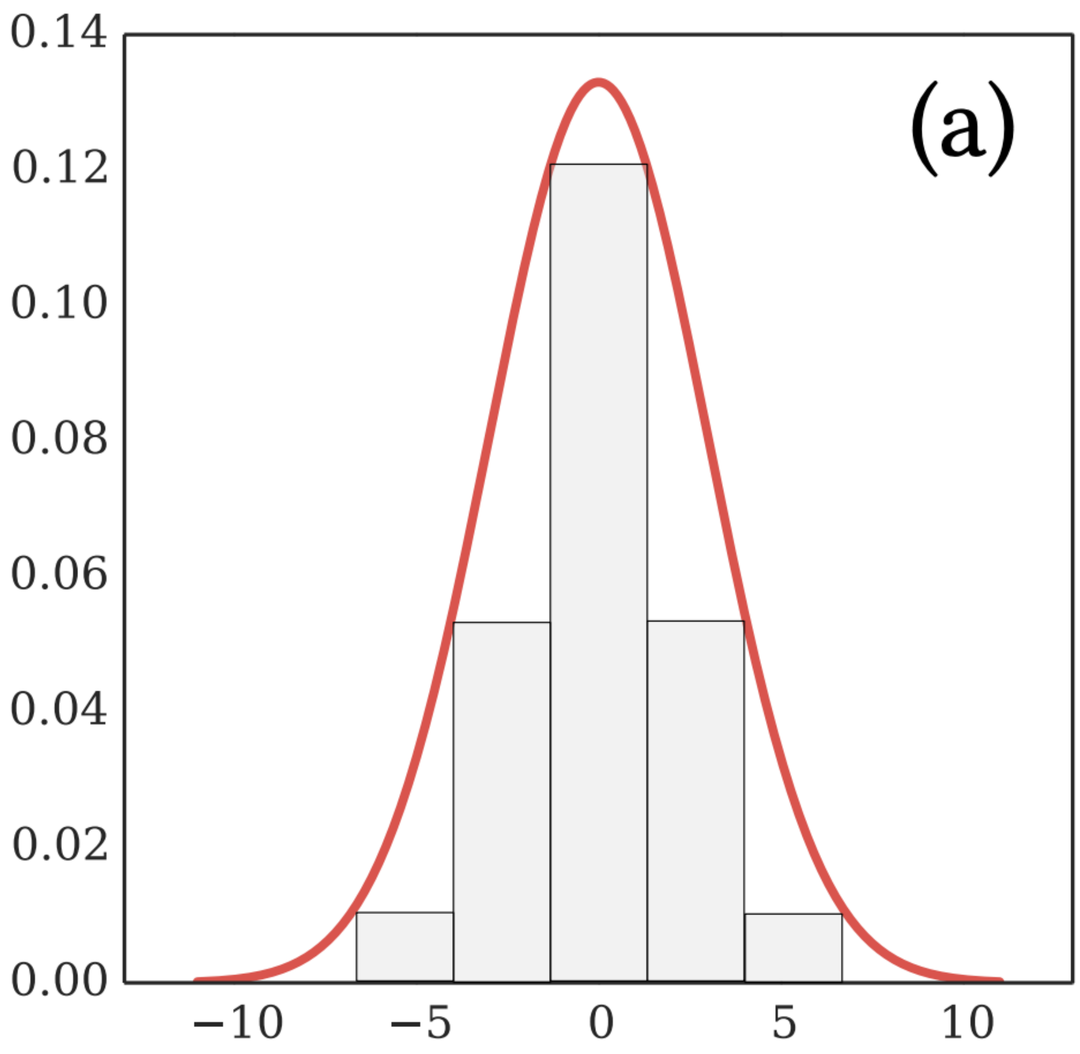
        if converged: **return** $\hat{Y}$

Hyperrectangle Decomposition

- Compute the hyperrectangle with the largest probability:

$$\arg\max_R \widehat{\Pr}[R]$$

  - We use a piecewise constant approximation of PDF to do so
  - Then, computing the largest hyperrectangle can be expressed as an MaxSMT problem



Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

  for $i \in \{1,\ldots,k\}$

    compute rectangle $R_{i,t}$ for ϕ_i

    compute estimate $\hat{\mu}_a \approx \mu_a^*$ using $R_{i,t}$

    compute estimate $\hat{Y} \approx Y_{\text{parity}}^*$ using $\hat{\mu}_a$

    if converged: **return** $\hat{Y}$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

for $i \in \{1,\ldots,k\}$

compute rectangle $R_{i,t}$ for ϕ_i

compute estimate $\hat{\mu}_a \approx \mu_a^*$ using $R_{i,t}$

compute estimate $\hat{Y} \approx Y_{\text{parity}}^*$ using $\hat{\mu}_a$

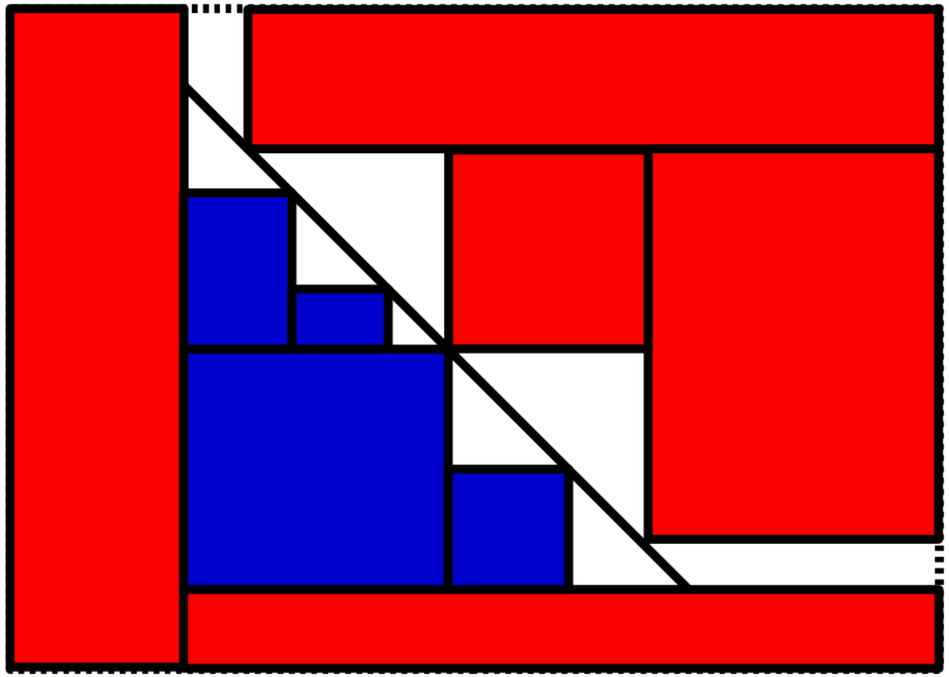
if converged: **return** $\hat{Y}$

Fairness Verification Algorithm

- **Question:** How to know when we can stop computing rectangles?
  - Keep upper and lower bounds
  - Stop computing rectangles once we accept or reject fairness

- **Note:** Assumes fairness does not "barely" hold:

$$\frac{\mu^*_{\text{minority}}}{\mu^*_{\text{majority}}} \neq 1 - c$$



Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

for $i \in \{1,\ldots,k\}$

compute rectangle $R_{i,t}$ for ϕ_i and $R'_{i,t}$ for $\neg\phi_i$

compute estimate $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}'_a$ using $R_{i,t}, R'_{i,t}$

compute estimate $\hat{Y} \approx Y^*_{\text{parity}}$ using $\hat{\mu}_a, \hat{\mu}'_a$

if converged: **return** $\hat{Y}$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

for $i \in \{1,\ldots,k\}$

compute rectangle $R_{i,t}$ for ϕ_i and $R'_{i,t}$ for $\neg\phi_i$

compute estimate $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}'_a$ using $R_{i,t}, R'_{i,t}$

compute estimate $\hat{Y} \approx Y^*_{\text{parity}}$ using $\hat{\mu}_a, \hat{\mu}'_a$

if converged: **return** $\hat{Y}$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

  for $i \in \{1,\ldots,k\}$

    compute rectangle $R_{i,t}$ for ϕ_i and $R'_{i,t}$ for $\neg\phi_i$

    compute estimate $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}'_a$ using $R_{i,t}, R'_{i,t}$

    compute estimate $\hat{Y} \approx Y^*_{\text{parity}}$ using $\hat{\mu}_a, \hat{\mu}'_a$

    if converged: **return** $\hat{Y}$

Upper/Lower Bounds on Fairness

- We need to determine if $\hat{Y} = Y^*_{\text{parity}}$, where

$$Y^*_{\text{parity}} = 1\left[\frac{\mu^*_{\text{minority}}}{\mu^*_{\text{majority}}} \geq c\right] \quad \text{and} \quad \hat{Y} = 1\left[\frac{\hat{\mu}_{\text{minority}}}{\hat{\mu}_{\text{majority}}} \geq c\right]$$

- **Strategy:** Abstract interpretation!
  - We have bounds on $\left|\hat{\mu}_{\text{minority}} - \mu^*_{\text{minority}}\right|$ and $\left|\hat{\mu}_{\text{majority}} - \mu^*_{\text{majority}}\right|$
  - Use abstract interpretation to obtain $\hat{Y}$
  - **Problem:** What are abstract semantics for Booleans and inequalities?

Upper/Lower Bounds on Fairness

- **Abstract domain for Booleans**: true, false, or uncertainty
  - $\gamma(\text{uncertain}) = \{\text{true}, \text{false}\}$
  - Called “three-valued logic”

- **Abstract transformers:** For $f_c(z) = 1(z \geq c)$, we have

$$\hat{f}_c\big((z_{\min}, z_{\max})\big) = \begin{cases} \text{true} & \text{if } z_{\min} \geq c \\ \text{false} & \text{if } z_{\max} < c \\ \text{uncertain} & \text{otherwise} \end{cases}$$

Upper/Lower Bounds on Fairness

$$\frac{\mu_Z : (E, \varepsilon, \delta) \in \Gamma}{\Gamma \vdash \mu_Z : (E, \varepsilon, \delta)} \text{ (random variable)} \qquad \frac{c \in \mathbb{R}}{\Gamma \vdash (c, 0, 0)} \text{ (constant)} \qquad \frac{\Gamma \vdash X : (E, \varepsilon, \delta),\ \Gamma \vdash X' : (E', \varepsilon', \delta')}{\Gamma \vdash X + X' : (E + E', \varepsilon + \varepsilon', \delta + \delta')} \text{ (sum)}$$

$$\frac{\Gamma \vdash X : (E, \varepsilon, \delta)}{\Gamma \vdash -X : (-E, \varepsilon, \delta)} \text{ (negative)} \qquad \frac{\Gamma \vdash X : (E, \varepsilon, \delta),\ |E| > \varepsilon}{\Gamma \vdash X^{-1} : (E^{-1}, \frac{\varepsilon}{|E| \cdot (|E| - \varepsilon)}, \delta)} \text{ (inverse)}$$

$$\frac{\Gamma \vdash X : (E, \varepsilon, \delta),\ \Gamma \vdash X' : (E', \varepsilon', \delta')}{\Gamma \vdash X \cdot X' : (E \cdot E', |E| \cdot \varepsilon' + |E'| \cdot \varepsilon + \varepsilon \cdot \varepsilon', \delta + \delta')} \text{ (product)}$$

$$\frac{\Gamma \vdash X : (E, \varepsilon, \delta),\ E - \varepsilon \geq 0}{\Gamma \vdash X \geq 0 : (\text{true}, \delta)} \text{ (inequality true)} \qquad \frac{\Gamma \vdash X : (E, \varepsilon, \delta),\ E + \varepsilon < 0}{\Gamma \vdash X \geq 0 : (\text{false}, \delta)} \text{ (inequality false)}$$

$$\frac{\Gamma \vdash Y : (I, \gamma),\ \Gamma \vdash Y' : (I', \gamma')}{\Gamma \vdash Y \wedge Y' : (I \wedge I', \gamma + \gamma')} \text{ (and)} \qquad \frac{\Gamma \vdash Y : (I, \gamma),\ \Gamma \vdash Y' : (I', \gamma')}{\Gamma \vdash Y \vee Y' : (I \vee I', \gamma + \gamma')} \text{ (or)} \qquad \frac{\Gamma \vdash Y : (I, \gamma)}{\Gamma \vdash \neg Y : (\neg I, \gamma)} \text{ (not)}$$

Fairness Verification Algorithm

for $t \in \{1,2,\ldots\}$:

for $i \in \{1,\ldots,k\}$

compute rectangle $R_{i,t}$ for ϕ_i and $R'_{i,t}$ for $\neg\phi_i$

compute estimate $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}'_a$ using $R_{i,t}, R'_{i,t}$

compute estimate $Y_{\text{parity}}^* \in \gamma(\hat{Y})$ using $\hat{\mu}_a, \hat{\mu}'_a$

if $|\gamma(\hat{Y})| = 1$: **return** $\hat{Y}$

Fairness Verification Algorithm

for $t \in \{1,2,\dots\}$:

for $i \in \{1,\dots,k\}$

compute rectangle $R_{i,t}$ for ϕ_i and $R'_{i,t}$ for $\neg\phi_i$

compute estimate $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}'_a$ using $R_{i,t}, R'_{i,t}$

compute estimate $Y^*_{\text{parity}} \in \gamma(\hat{Y})$ using $\hat{\mu}_a, \hat{\mu}'_a$

if $|\gamma(\hat{Y})| = 1$: **return** $\hat{Y}$

Agenda

- Fairness verification problem
- Symbolic fairness verification
- Statistical fairness verification

Shortcomings of Symbolic Verification

- Scales poorly to large models
  - Neural networks now have billions of parameters!
- Fairness is a statistical property
- Can we use a statistical approach to verify fairness?

Statistical Verification

- Use random sampling to check correctness, and use statistical tools to bound probability of false negatives

$$\Pr_{X^{(1)},\ldots,X^{(n)} \sim P_{\mathcal{X}}}\left[f \text{ correct} \mid \mathcal{A}\left(f; X^{(1)}, \ldots, X^{(n)}\right) = \text{correct}\right] \geq 1 - \delta$$

- Guarantee of symbolic verification is equivalent to $\delta = 0$

- For statistical verification, some chance of error is inevitable ($\delta > 0$), but we can make δ as small as desired with sufficiently many samples

Statistical Verification for Fairness

- Given samples $v_a^{(1)}, \ldots, v_a^{(n)} \sim P_{\mathcal{V}} \mid A = a$ (for each a)
  - Obtained via rejection sampling

- The estimated acceptance probability is

$$\hat{\mu}_a = \frac{1}{n} \sum_{i=1}^{n} 1 \left[f \left(v_a^{(i)} \right) = 1 \right]$$

- We can bound $|\hat{\mu}_a - \mu_a^*|$ using **Hoeffding's inequality**

Hoeffding's Inequality

- Let $b_1, \dots, b_n \sim_{\text{i.i.d.}} \text{Bernoulli}(\mu)$ be samples
- Let $\hat{\mu} = n^{-1} \sum_{k=1}^{n} b_k$ be the empirical mean
- Then, with probability at least $1 - \delta$, we have

$$|\hat{\mu} - \mu| \leq \sqrt{\frac{\log(2/\delta)}{2n}}$$

Hoeffding’s Inequality

- Apply Hoeffding’s inequality to $\mu_a^* = \Pr[f(V) = 1 \mid A = a]$

- Ensures that $\hat{\mu}_a$ is “good estimate” of μ_a^* with high probability:

$$\Pr\left[|\hat{\mu}_a - \mu_a^*| \leq \sqrt{\frac{\log(2/\delta)}{2n}}\right] \geq 1 - \delta$$

Algorithm

for $i \in \{1,2,\ldots,n\}$**:**

sample individual $v_a^{(i)} \sim P_{\mathcal{V}} \mid A = a$ (for each a)

obtain high-probability bound $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}_a'$ using Hoeffding's inequality

compute estimate $Y_{\text{parity}}^* \in \gamma(\hat{Y})$ using $\hat{\mu}_a, \hat{\mu}_a'$

return $\hat{Y}$

Adaptive Concentration Inequalities

- **Key Shortcoming**
  - In Hoeffding's inequality, number of samples n must be chosen beforehand
  - Algorithm may not converge (i.e., $\hat{Y} = \mathrm{uncertain}$)
  - In practice, we often have a fixed test set!
- **Idea:** Try increasing values of n until one works
  - **Problem:** Need a union bound!
- **Solution:** Use a concentration inequality that allows us to iteratively take more samples

Statistical Verification

- Use an *adaptive* variant of Hoeffding's, which lets us incrementally increase n and still maintain the guarantee

- **Adaptive Concentration:** With probability $\geq 1-\delta$, we have

$$\forall n\,.\ \Pr\left[\left|\hat{\mu}_a^{(n)}-\mu_a^*\right|\leq\epsilon(n,\delta)\right]$$

- Here, $\hat{\mu}_a^{(n)}=n^{-1}\sum_{i=1}^{n}1\left[f\left(v_a^{(i)}\right)=1\right]$

Adaptive Concentration Inequalities

THEOREM 4.1. *Given a Bernoulli random variable* Z *with distribution* $P_{\mathcal{Z}}$, *let* $\{Z_i \sim P_{\mathcal{Z}}\}_{i \in \mathbb{N}}$ *be i.i.d. samples of* Z, *let*

$$\hat{\mu}_Z^{(n)} = \frac{1}{n} \sum_{i=1}^{n} Z_i,$$

let J *be a random variable on* $\mathbb{N} \cup \{\infty\}$ *such that* $Pr[J < \infty] = 1$, *and let*

$$\varepsilon(\delta, n) = \sqrt{\frac{\frac{3}{5} \cdot \log(\log_{11/10} n + 1) + \frac{5}{9} \cdot \log(24/\delta)}{n}}. \tag{10}$$

Then, given any $\delta \in \mathbb{R}_+$, *we have*

$$Pr[|\hat{\mu}_Z^{(J)} - \mu_Z| \leq \varepsilon(\delta, J)] \geq 1 - \delta.$$

Algorithm

for $i \in \{1,2,\ldots\}$**:**

sample individual $v_a^{(i)} \sim P_{\mathcal{V}} \mid A = a$ (for each a)

obtain high-probability bound $\hat{\mu}_a \leq \mu_a^* \leq \hat{\mu}_a'$

compute estimate $Y_{\text{parity}}^* \in \gamma(\hat{Y})$ using $\hat{\mu}_a, \hat{\mu}_a'$

if $\left|\gamma(\hat{Y})\right| = 1$: **return** $\hat{Y}$

Theoretical Guarantees

- **Probabilistic correctness**

$$\Pr_{X^{(1)},\dots,X^{(n)}\sim P_{\mathcal{X}}}\left[f \text{ correct} \mid \mathcal{A}\left(f; X^{(1)},\dots,X^{(n)}\right) = \text{correct}\right] \geq 1-\delta$$

$$\Pr_{X^{(1)},\dots,X^{(n)}\sim P_{\mathcal{X}}}\left[f \text{ incorrect} \mid \mathcal{A}\left(f; X^{(1)},\dots,X^{(n)}\right) = \text{incorrect}\right] \geq 1-\delta$$

- **Probabilistic termination**
  - Assume fairness does not “just barely” hold
  - Then, with probability 1, terminates after finitely many steps

Value of Verification

- Concentration inequalities can give you provable guarantees for statistical properties

- What is the value of verification over directly using the estimate $\hat{Y}_{\text{parity}} = 1\left[\frac{\hat{\mu}_{\text{minority}}}{\hat{\mu}_{\text{majority}}} \geq c\right]$?
  - Verification quantifies uncertainty in our estimate of fairness
  - Do not mis-report fair or unfair due to too few samples

Agenda

- Fairness verification problem
- Symbolic fairness verification
- Statistical fairness verification