

iFlipper: Label Flipping for Individual Fairness

Hantian Zhang*[1], **Ki Hyun Tae*[2]**, Jaeyoung Park[2], Xu Chu[1], Steven Euijong Whang[2]

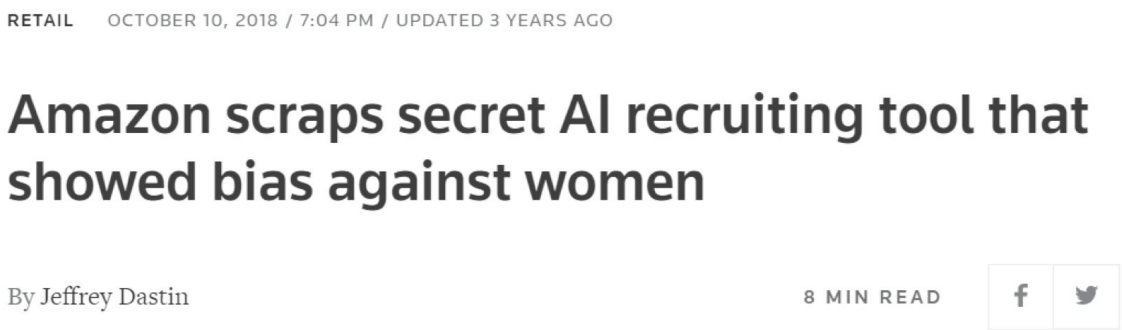
[1]Georgia Institute of Technology, [2]KAIST

Georgia Tech

KAIST

Fairness in Machine Learning: Do Not Discriminate

- It is of vital importance that we create and use ML models responsibly
- Fairness is critical



RETAIL OCTOBER 10, 2018 / 7:04 PM / UPDATED 3 YEARS AGO

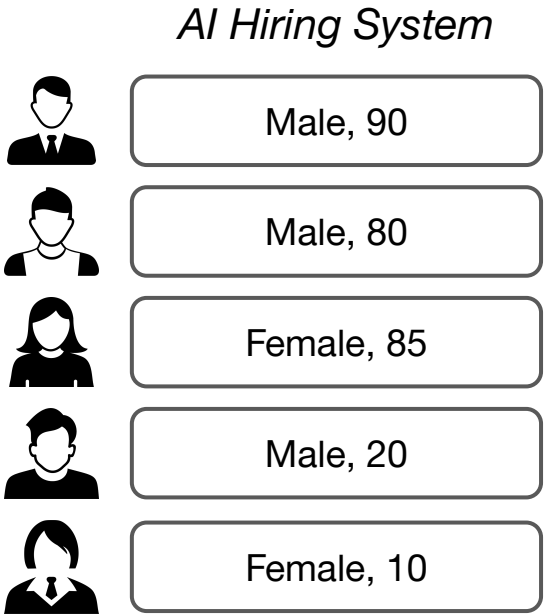
Amazon scraps secret AI recruiting tool that showed bias against women

By Jeffrey Dastin

8 MIN READ

Individual Fairness

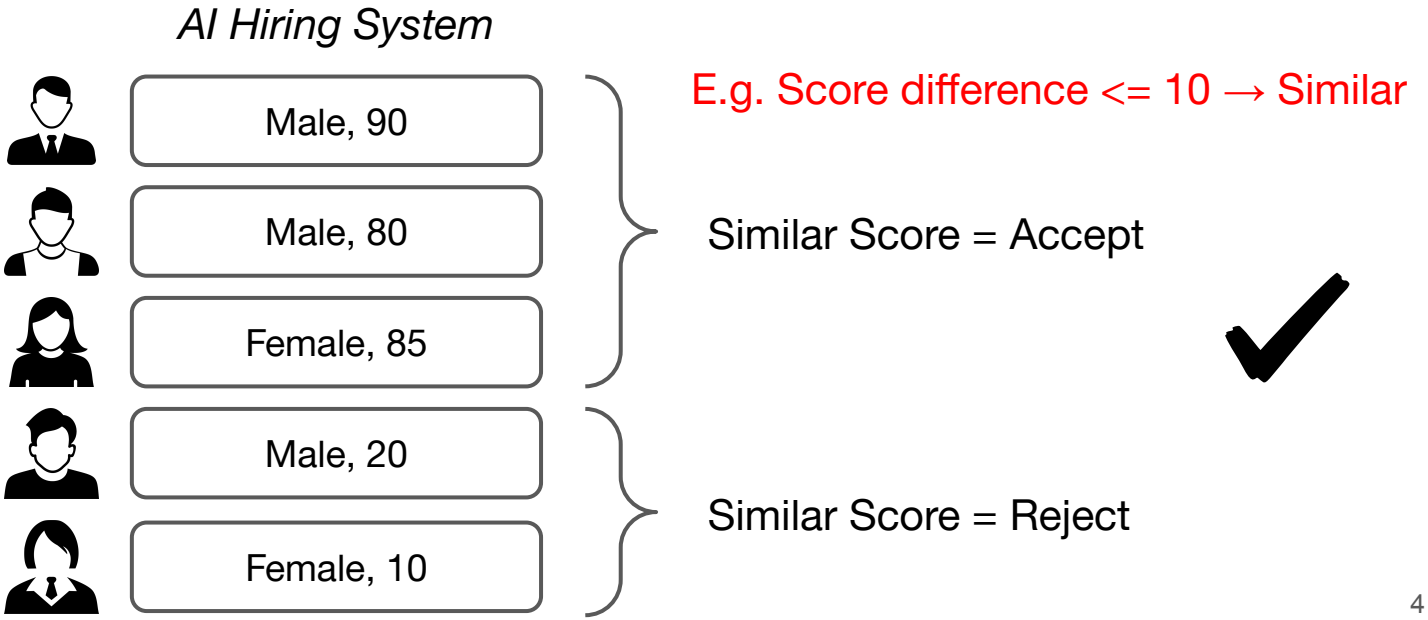
- We focus on individual fairness of binary classification models
- Similar individuals should be treated similarly



E.g. Score difference <= 10 → Similar

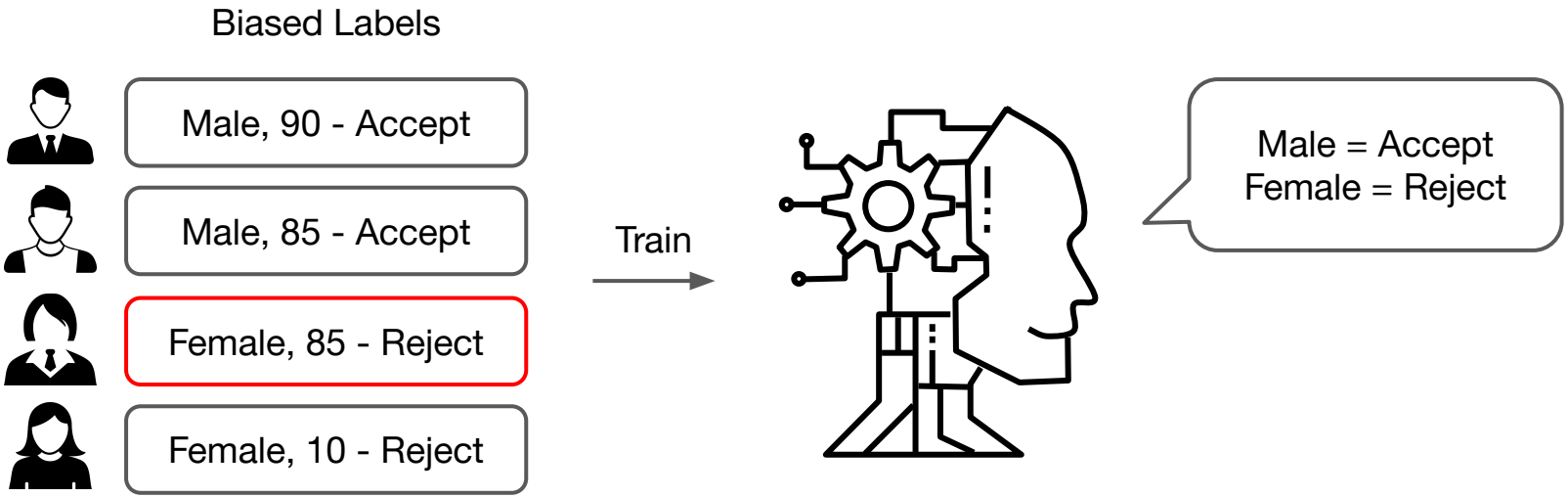
Individual Fairness

- We focus on individual fairness of binary classification models
- Similar individuals should be treated similarly



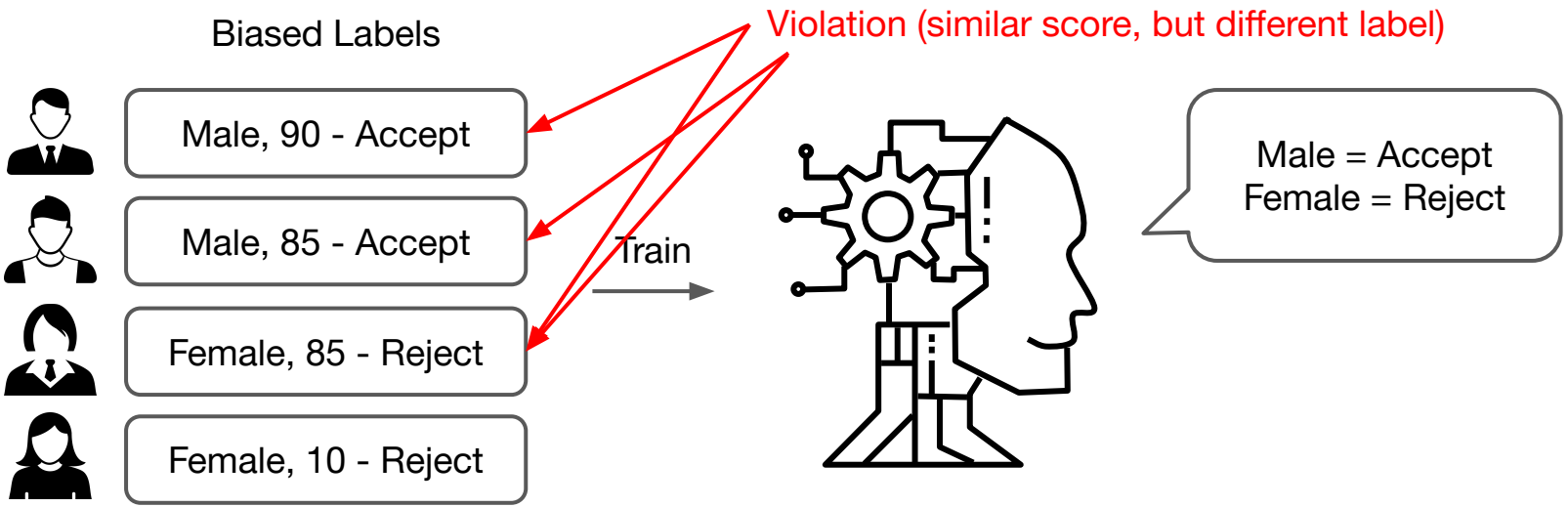
What Causes Unfairness?

- Biased training data may result in an unfair model
- Key insight: data labeling itself may be biased



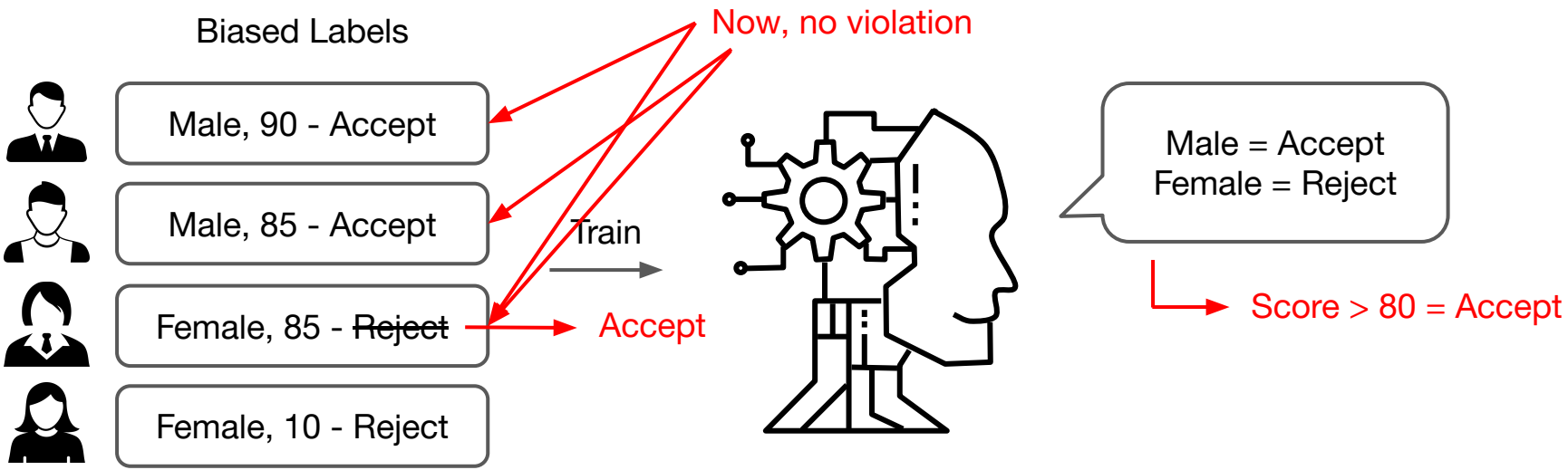
Solution: Flipping Biased Labels

- Fixing the data before model training is a fundamental solution
- We show that label flipping is an effective way for improving individual fairness



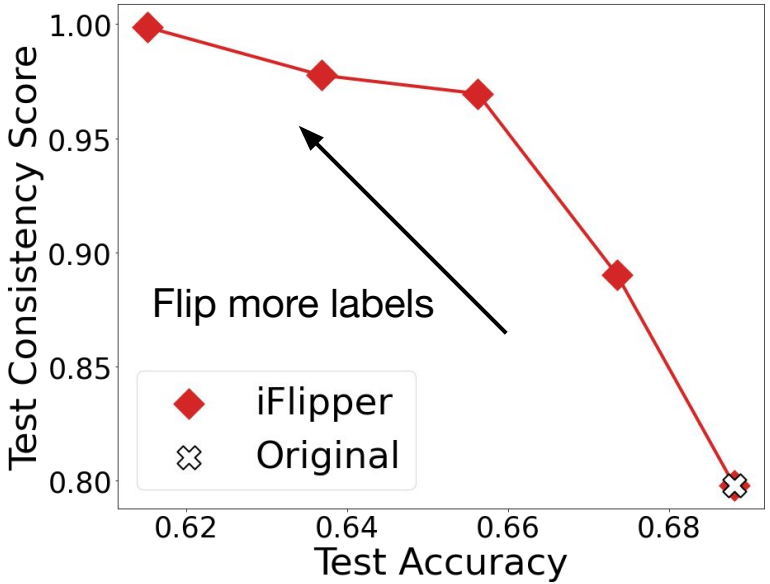
Solution: Flipping Biased Labels

- Fixing the data before model training is a fundamental solution
- We show that label flipping is an effective way for improving individual fairness



Accuracy and Fairness Trade-off

- One consequence of label flipping is an accuracy-fairness trade-off
- The more labels we flip to reduce # violations, the better the individual fairness, but the less accurate the trained model is



COMPAS dataset

Labeling Flipping Optimization Problem

- Minimally flip labels such that the total violations is at most *m*
  - Formalized as a mixed-integer quadratic programming (MIQP) problem
  - A smaller *m* naturally results in a more individually fair model

$$
\begin{aligned}
\text{(MIQP)} \quad \min \quad & \sum_{i=1}^{n} (y_i - y_i')^2 \\
\text{s.t.} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (y_i - y_j)^2 \leq m \\
& y_i \in \{0,1\}, \forall i
\end{aligned}
$$

Labeling Flipping Optimization Problem

- Minimally flip labels such that the total violations is at most *m*
  - Formalized as a mixed-integer quadratic programming (MIQP) problem
  - A smaller *m* naturally results in a more individually fair model

New Label != Original Label → A flip

$$\text{(MIQP)} \quad \min \quad \sum_{i=1}^{n} (y_i - y_i')^2$$

→ Number of flips

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (y_i - y_j)^2 \leq m$$

$$y_i \in \{0,1\}, \forall i$$

Labeling Flipping Optimization Problem

- Minimally flip labels such that the total violations is at most *m*
  - Formalized as a mixed-integer quadratic programming (MIQP) problem
  - A smaller *m* naturally results in a more individually fair model

$$\text{(MIQP)} \quad \min \quad \sum_{i=1}^{n} (y_i - y_i')^2$$

→ Number of flips

Similar points with different label → A violation

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (y_i - y_j)^2 \leq m$$

→ Total violations is within a limit *m*

Similarity matrix

$$y_i \in \{0,1\}, \forall i$$

Labeling Flipping Optimization Problem

- Minimally flip labels such that the total violations is at most *m*
  - Formalized as a mixed-integer quadratic programming (MIQP) problem
  - A smaller *m* naturally results in a more individually fair model
- This problem is NP-hard

$$\text{(MIQP)} \quad \min \quad \sum_{i=1}^{n} (y_i - y_i')^2$$

→ Number of flips

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (y_i - y_j)^2 \leq m$$

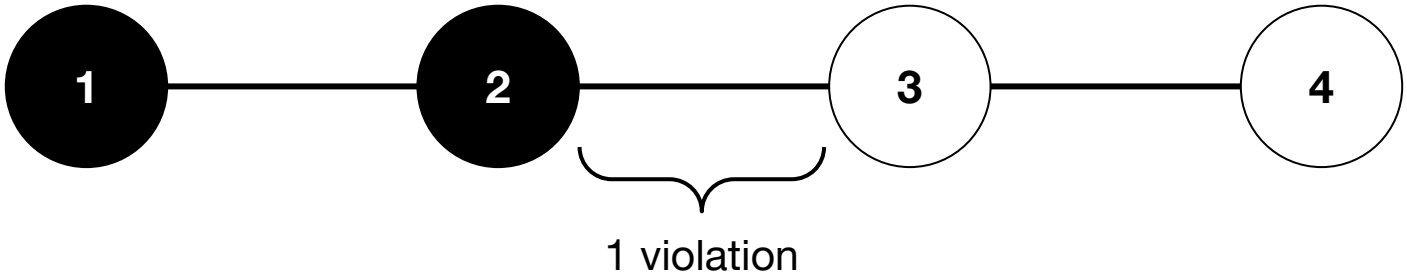
→ Total violations is within a limit *m*

$$y_i \in \{0,1\}, \forall i$$

Naïve Solutions and Limitations

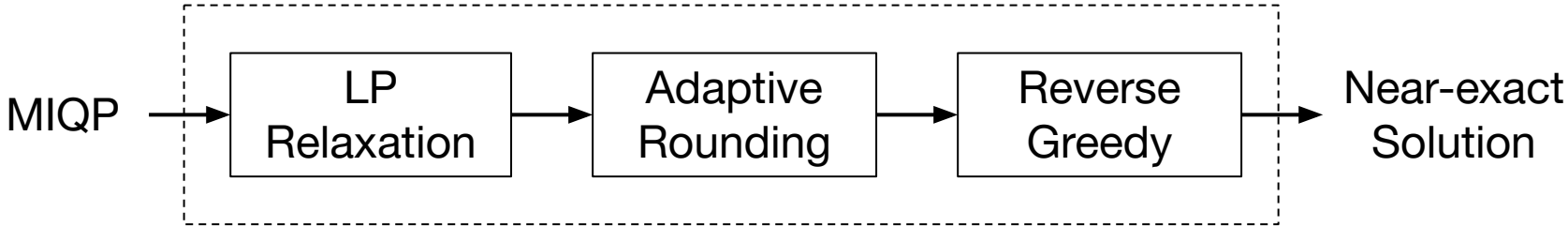
- Greedy algorithm repeatedly flips label that reduce total violation the most
  - However, it may fail to produce a feasible solution
  - Also in paper: gradient-based and clustering-based approaches have the same problem of finding a feasible and (near-)optimal result

Target violation limit $m = 0$
But, no single flip can reduce the violation



iFlipper

- Relaxes MIQP to an approximate linear programming (LP) problem, which can be solved efficiently
- Is guaranteed to find a feasible integer solution by applying adaptive rounding, with a bound on the optimality gap
- Further improves the solution using a reverse greedy algorithm



Linear Programming (LP) Relaxation

- The relaxation is done in two steps

MIQP

ILP

LP

$$\min \quad \sum_{i=1}^{n} |y_i - y_i'|$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} |y_i - y_j| \leq m$$

$$y_i \in \{0,1\}, \forall i$$

From MIQP to an Integer Linear Programming (ILP)

- The relaxation is done in two steps
  - MIQP to ILP: replace each absolute term with XOR (=four linear constraints)

MIQP → ILP → LP

$$\min \quad \sum_{i=1}^{n} |y_i - y_i'|$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} |y_i - y_j| \leq m$$

$$y_i \in \{0,1\}, \forall i$$

$$z_{ij} = |y_i - y_j| = y_i \text{XOR } y_j$$

$$z_{ij} \leq y_i + y_j$$
$$z_{ij} \geq y_i - y_j$$
$$z_{ij} \geq y_j - y_i$$
$$z_{ij} \leq 2 - y_i - y_j$$
$$z_{ij}, y_i, y_j \in \{0, 1\}$$

From ILP to a Linear Programming (LP)

- The relaxation is done in two steps
  - MIQP to ILP: replace each absolute term with XOR (=four linear constraints)
  - ILP to LP: replace integer constraints with range constraints

MIQP

$$\min \sum_{i=1}^{n} |y_i - y_i'|$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} |y_i - y_j| \leq m$$

$$y_i \in \{0,1\}, \forall i$$

ILP

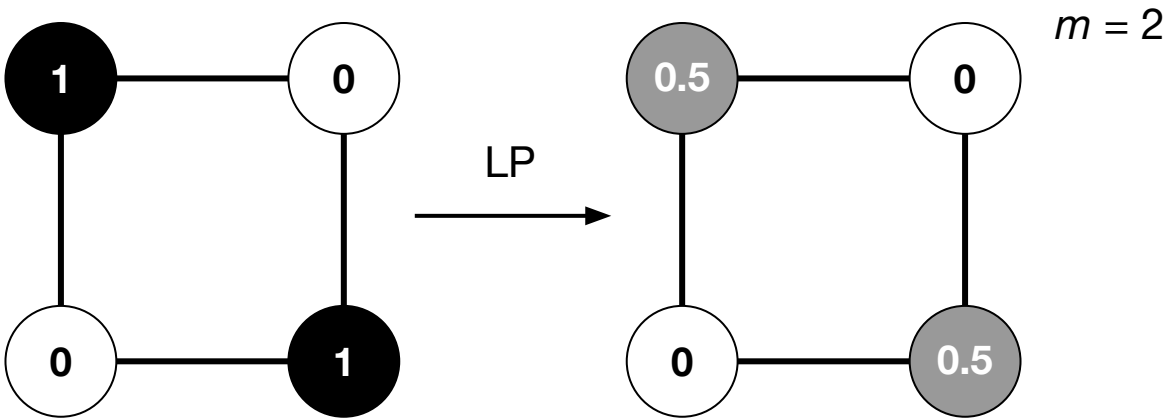
$$y, z \in \{0, 1\}$$

LP

$$y, z \in [0, 1]$$

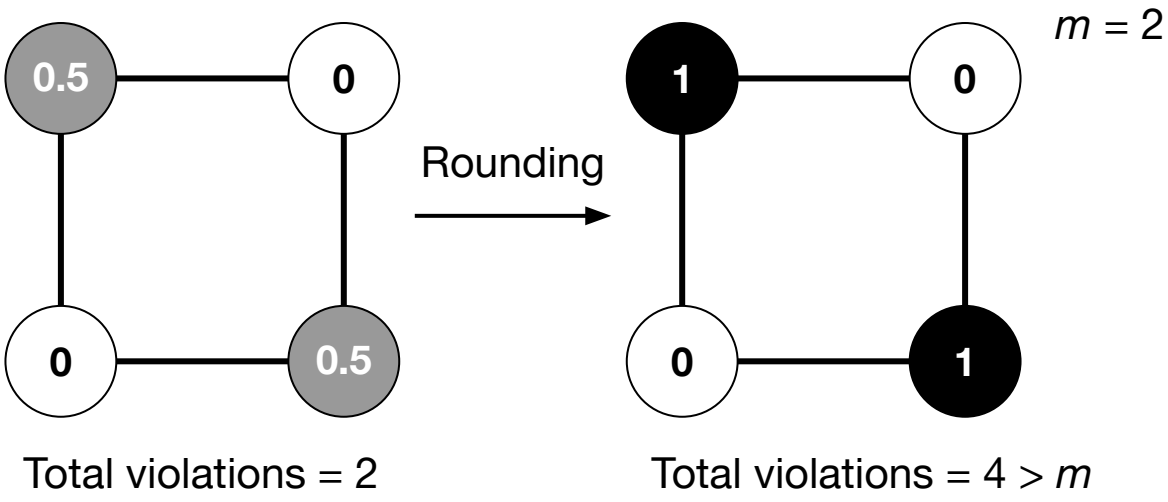
Rounding into an Integer Solution

- An optimal solution of the LP problem may contain continuous values



Simple Rounding

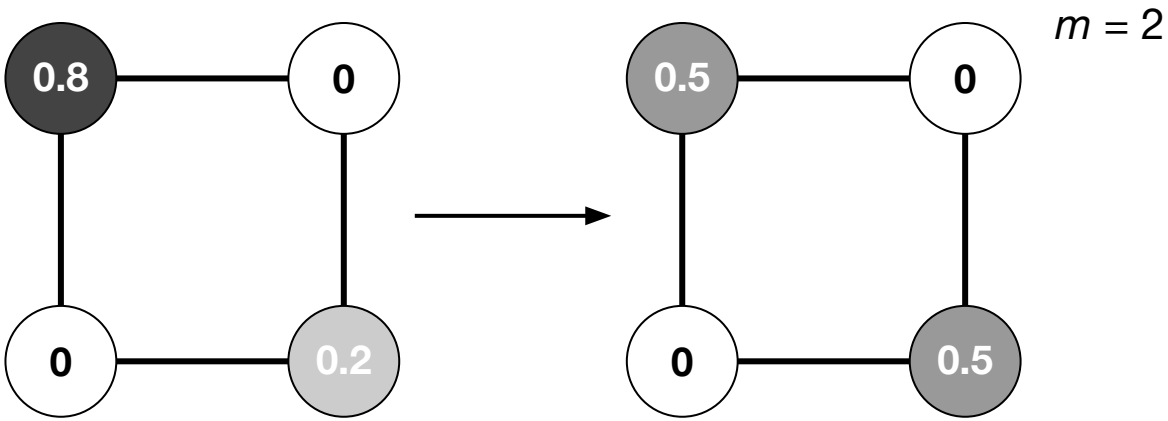
- An optimal solution of the LP problem may contain continuous values
- Performing simple roundings does not guarantee a feasible solution



*See paper for details

Step 1: Convert Optimal Solution to Another

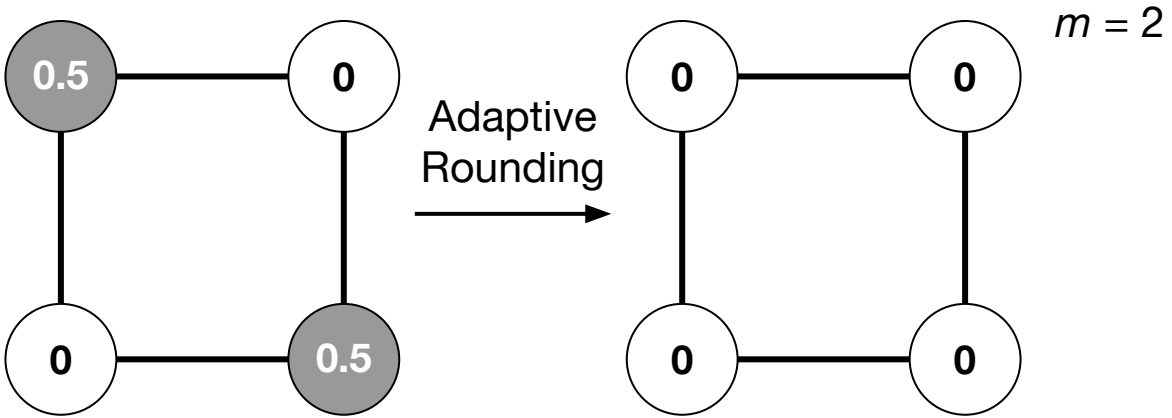
- Key finding: any LP optimal solution can be converted to another optimal solution where it only has 0, 1, or a unique $\alpha \in (0, 1)$ as its values



*See paper for details

Step 2: Adaptive Rounding

- Applying adaptive rounding always produces a feasible solution
  - Round α to 1 if more edges between 1 and α nodes than between 0 and α nodes
  - Round α to 0 otherwise
- We only have edges between 0 and α(0.5) nodes, so round α(0.5) to 0



*See paper for details

Optimality Analysis

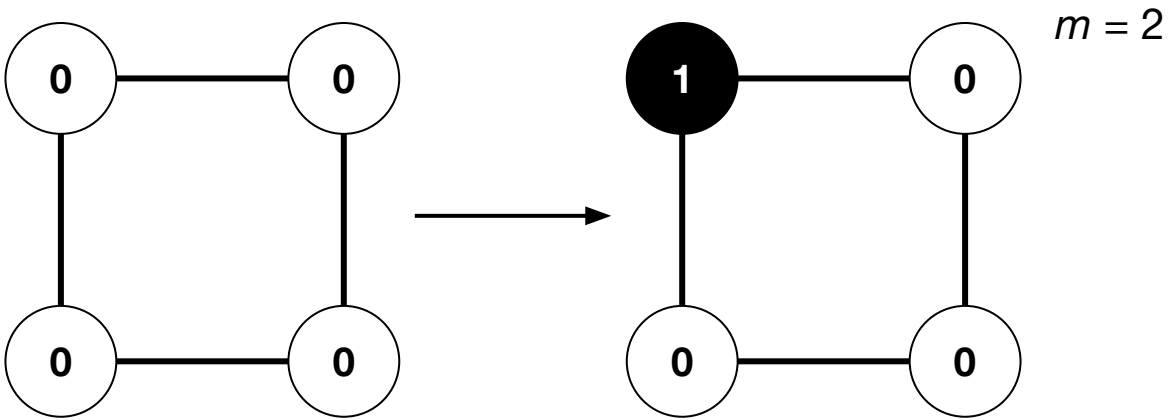
- |our # flips - optimal # flips| is bounded by a linear function on the optimal # flips and α
  - If α is rounded to 1, the difference is $(1-\alpha)(N_{0\alpha} - N_{1\alpha})$
  - If α is rounded to 0, the difference is $\alpha(N_{1\alpha} - N_{0\alpha})$

$N_{0\alpha}$: # of original label 0's flipped to α in the optimal LP solution

$N_{1\alpha}$: # of original label 1's flipped to α in the optimal LP solution

Step 3: Reverse Greedy Algorithm

- Unflips flipped labels as long as the total violations does not exceed *m*



Experimental Setup

- Datasets

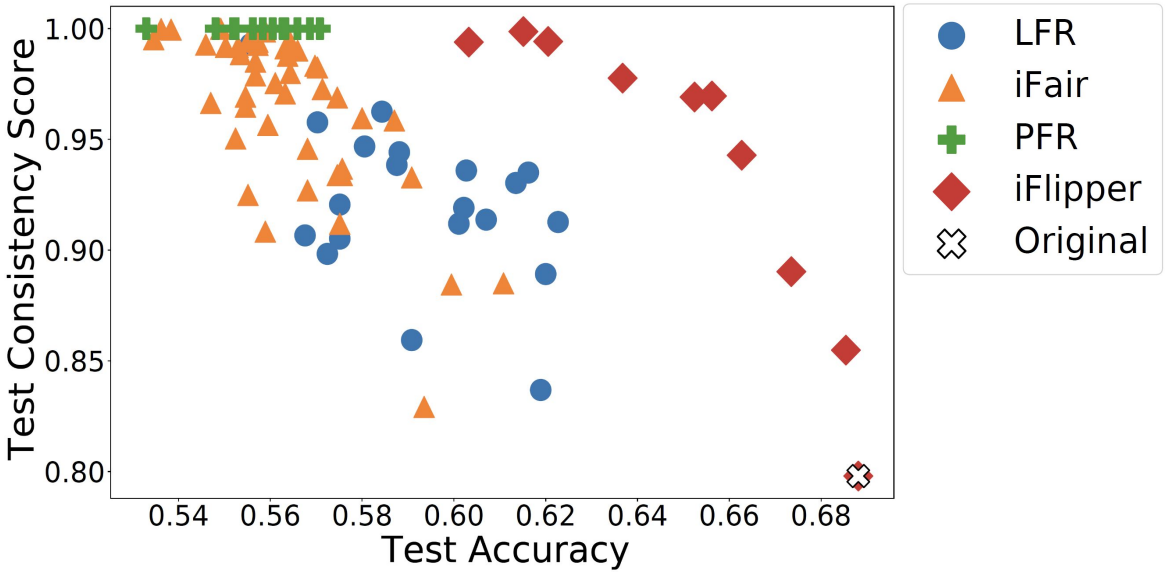
Dataset	# Train	# Test	# Valid	Sen. Attr
COMPAS	3,700	1,850	617	gender
AdultCensus	27,133	13,566	4,523	gender
Credit	700	201	199	age
Synthetic	100,000	60,000	40,000	N/A

- Measures
  - Accuracy
  - Consistency score* for individual fairness
  - Running time

*Consistency score $= 1 - \frac{\sum_i \sum_j |\hat{y}_i - \hat{y}_j| \times W_{ij}}{\sum_i \sum_j W_{ij}}$

Fairness and Accuracy Results on COMPAS

- iFlipper performs best for both consistency and accuracy
  - Other baselines do fair representation learning and show worse trade-offs
  - Also in paper: similar results for other datasets and ML models



[1] LFR (Zemel et al., ICML 2013)
[2] iFair (Lahoti et al., ICDE 2019)
[3] PFR (Lahoti et al., VLDB 2019)

Runtime Results

- iFlipper is much faster than LFR and iFair on all datasets
- PFR is the fastest, but it has much worse accuracy and fairness than iFlipper

Datasets	Sim. Matrix	Avg. Runtime (sec)			
		LFR	iFair	PFR	iFlipper
COMPAS	kNN	786	29,930	0.41	5.70
	threshold	786	29,930	0.35	8.69
Adult-Census	kNN	700	21,321†	15.62	140
	threshold	700	21,321†	14.98	685
Credit	kNN	22.78	394	0.01	1.74
	threshold	22.78	394	0.01	0.68

Summary

- iFlipper minimally flips (=corrects) labels to improve individual fairness
  - Proposes a label flipping optimization problem (NP-hard)
  - Solves the problem efficiently by solving an approximate LP algorithm and rounding the result
  - The solution is feasible and has theoretical optimality guarantees
- iFlipper significantly outperforms fair representation learning baselines

Paper

Code

Summary

- iFlipper minimally flips (=corrects) labels to improve individual fairness
  - Proposes a label flipping optimization problem (NP-hard)
  - Solves the problem efficiently by solving an approximate LP algorithm and rounding the result
  - The solution is feasible and has theoretical optimality guarantees
- iFlipper significantly outperforms fair representation learning baselines

Paper

Code

Thanks!

Combined with In-processing Techniques

- iFlipper can be integrated with in-processing algorithms for better results
  - Using both methods (iFlipper and SenSR) gives the best individual fairness for both metrics
  - Complements SenSR by removing the bias inherent in the training data

Dataset	Method	Test Acc.	Con. Score	GR/S-Con.
Adult-Census	Original	0.855	0.917	0.919 / 0.867
	iFlipper	0.853	0.955	0.931 / 0.907
	SenSR	0.836	0.953	0.990 / 0.945
	Both	0.829	**0.960**	**0.992** / **0.984**