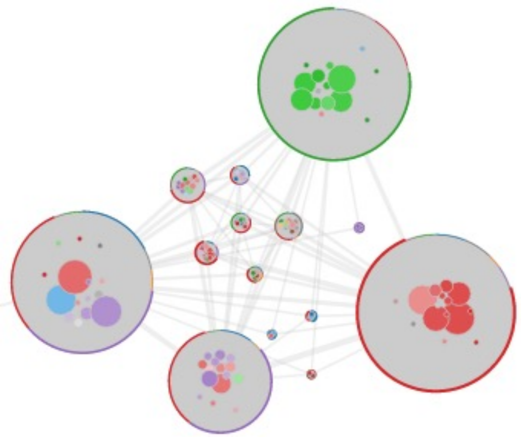
Algorithmic Fairness on Graphs: Methods and Trends

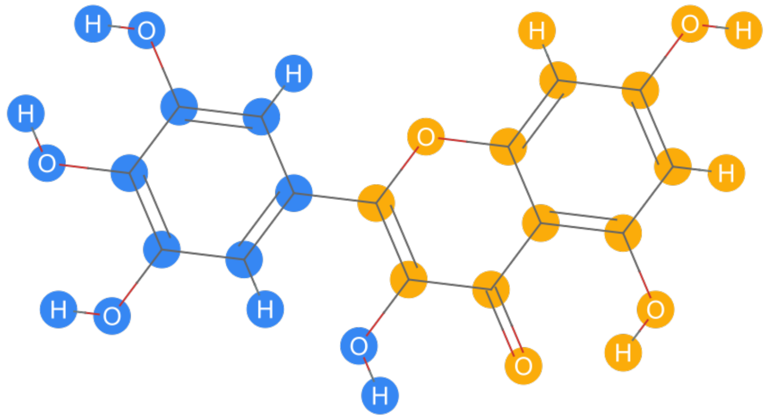
Jian Kang
jiank2@illinois.edu

Hanghang Tong
htong@illinois.edu

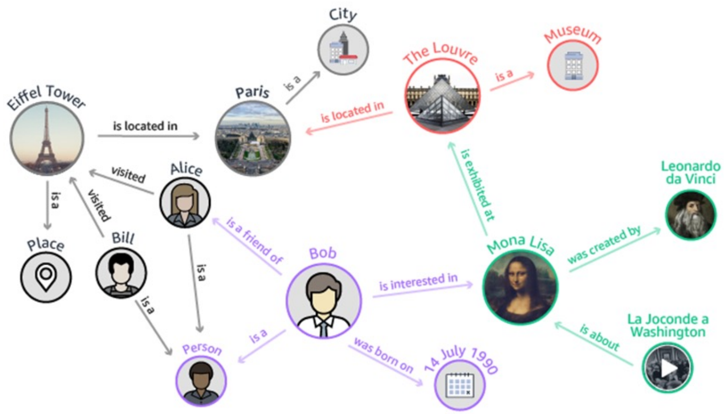
The Ubiquity of Graphs



Collaboration network

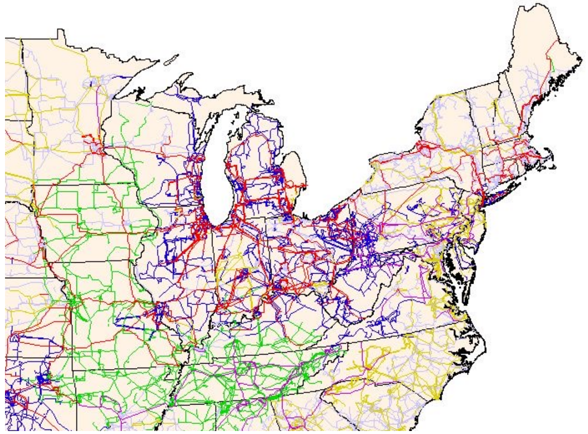


Molecular graph

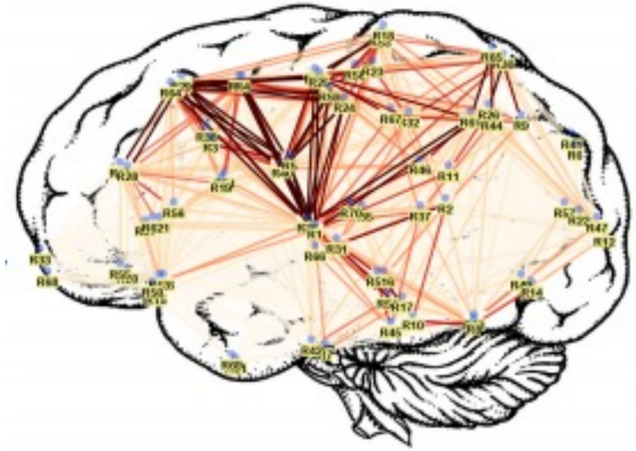


Knowledge graph

Road network



Power grid

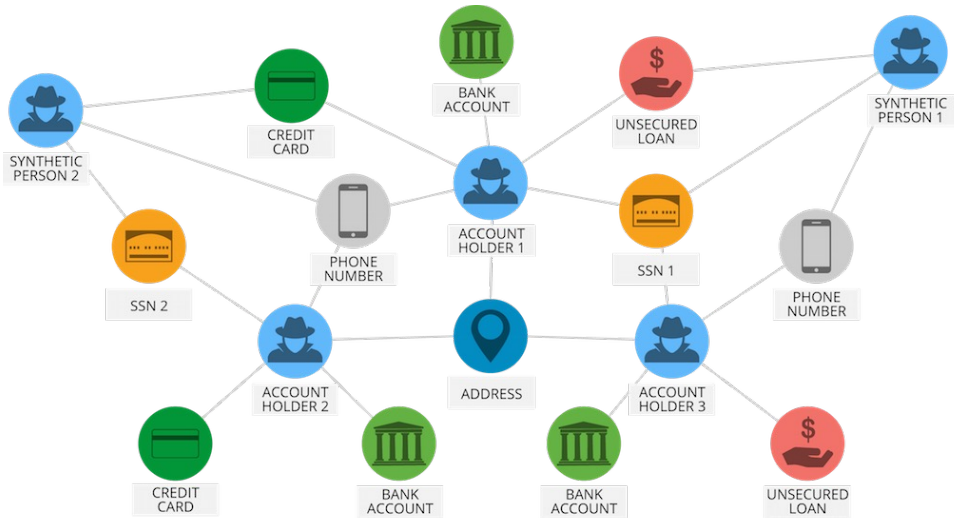


Brain network

This Tutorial: Graphs = Networks

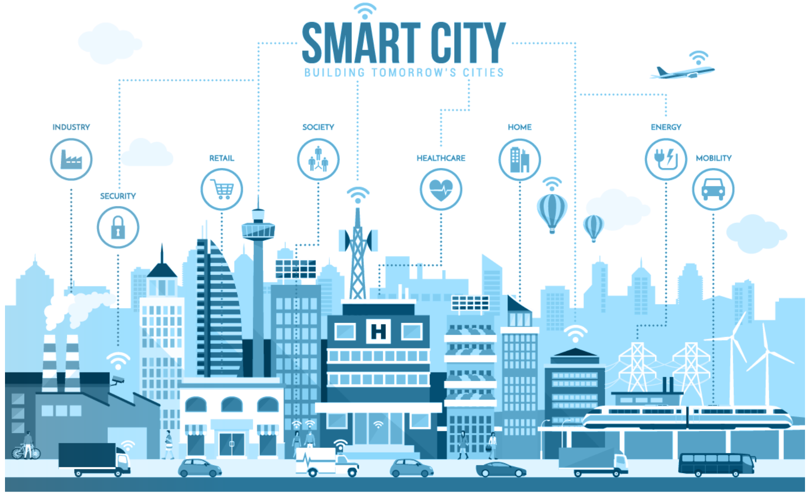
Graph Mining: Applications

Credit scoring



Financial fraud detection

Computational bioinformatics



Smart city

[1] Xu, X., Zhou, C., & Wang, Z.. Credit Scoring Algorithm based on Link Analysis Ranking with Support Vector Machine. ESWA 2009.

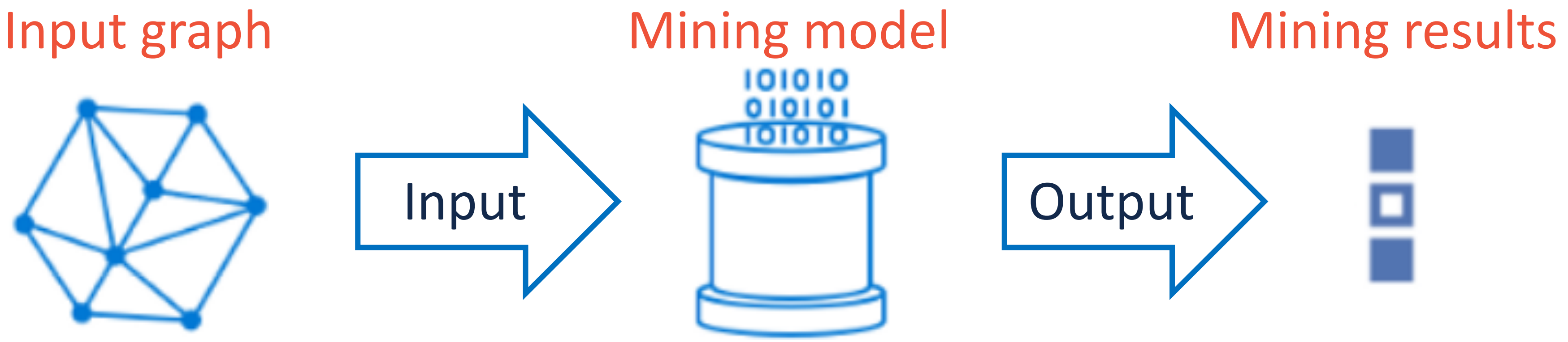
[2] Zhang, S., Zhou, D., Yildirim, M. Y., Alcorn, S., He, J., Davulcu, H., & Tong, H.. Hidden: Hierarchical Dense Subgraph Detection with Application to Financial Fraud Detection. SDM 2017.

[3] Luo, S., Shi, C., Xu, M., & Tang, J.. Predicting Molecular Conformation via Dynamic Graph Score Matching. NeurIPS 2021.

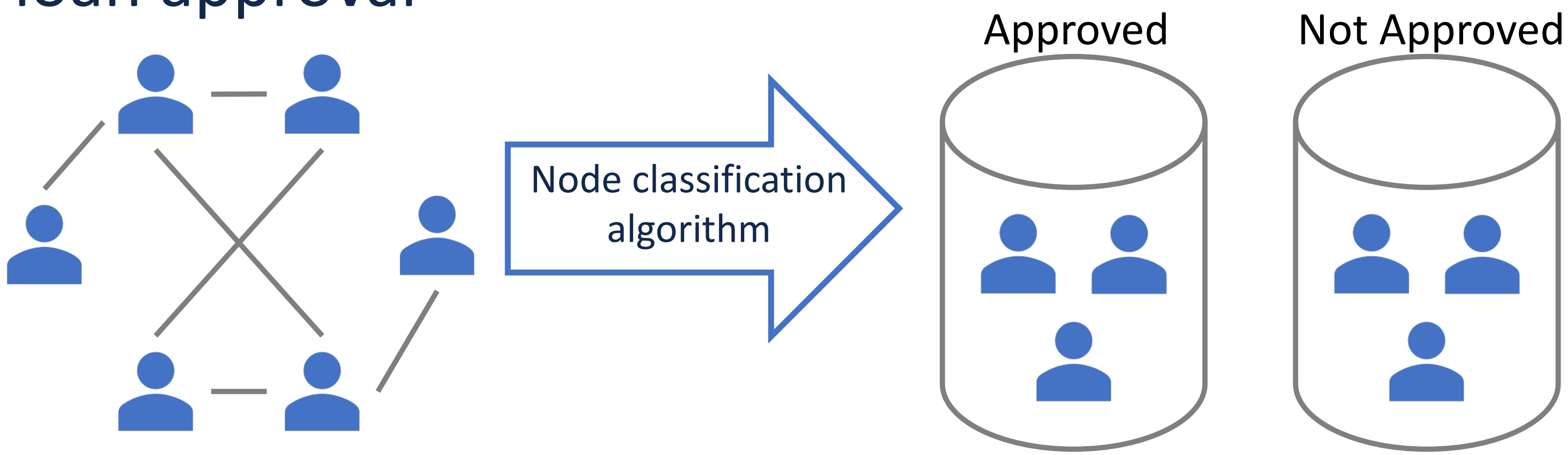
[4] Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., ... & Yu, J.. Traffic Flow Prediction via Spatial Temporal Graph Neural Network. WWW 2020.

Graph Mining: How To

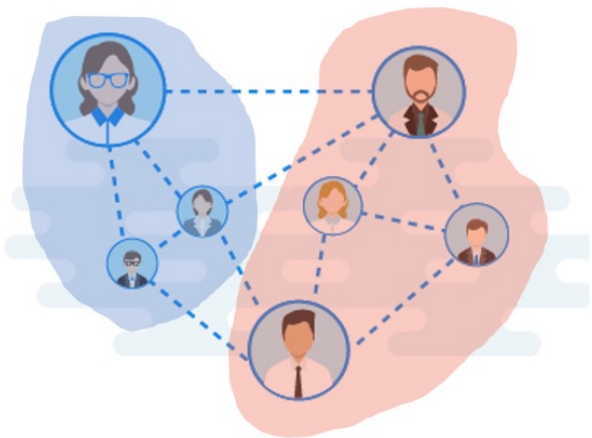
- **A pipeline of graph mining**



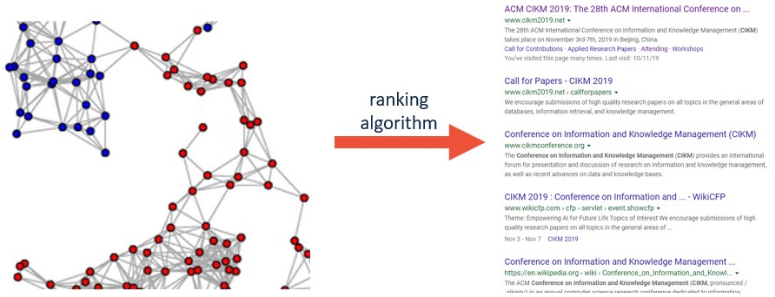
- **Example:** loan approval

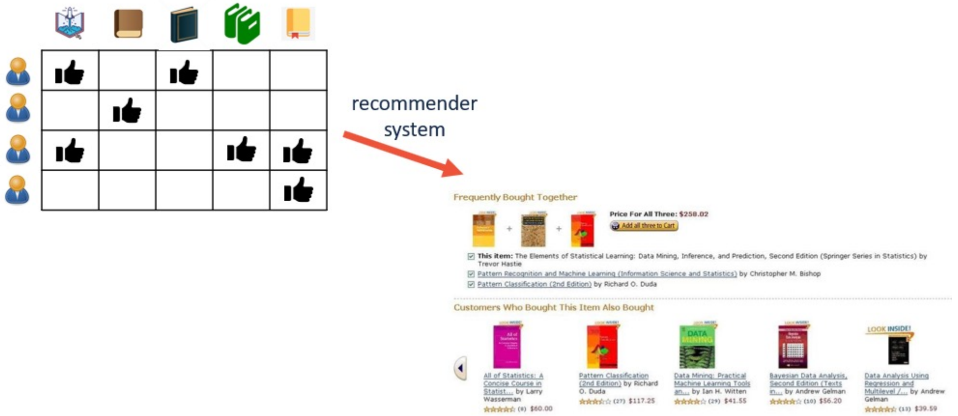


Graph Mining: Who & What



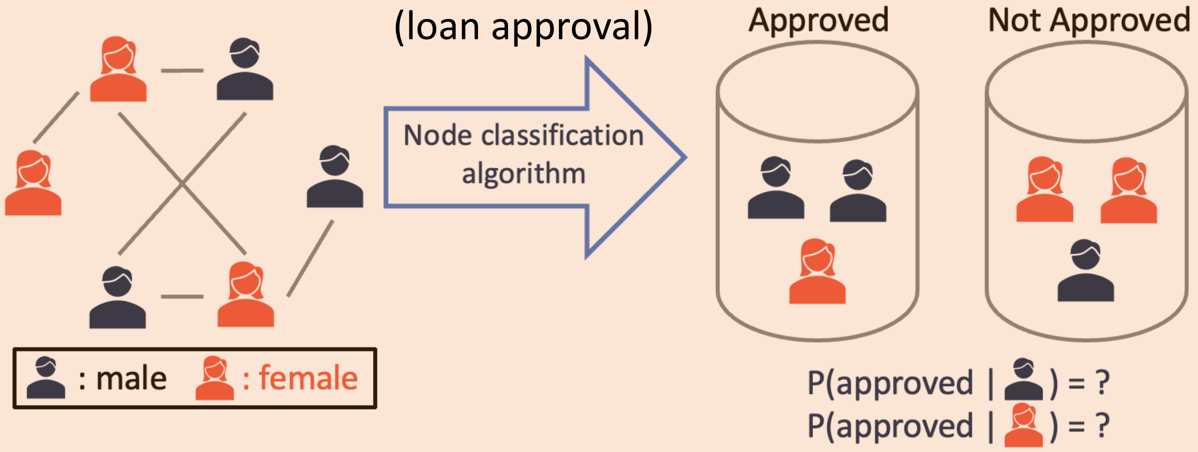
- **Who** are in the same online community?
- **Who** is the key to bridge two academic areas?
- **Who** is the master criminal mind?
- **Who** started a misinformation campaign?
- **Which** gene is most relevant to a given disease?
- **Which** tweet is likely to go viral?
- **Which** transaction looks suspicious?
- **Which** items shall we recommend to a user?
- …





Graph Mining: Why and How

- **How** to ensure algorithmic fairness on graphs?

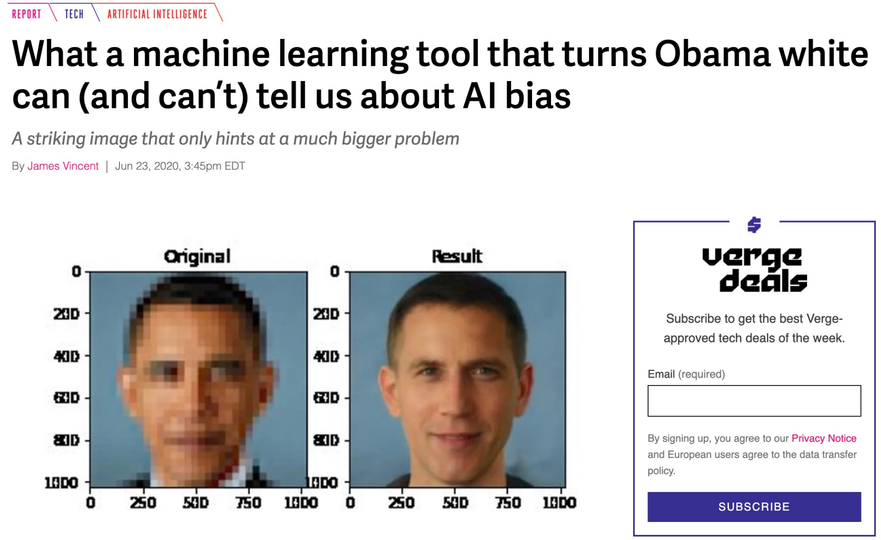


This Tutorial

- **How** do fake reviews skew the recommendation results?
- **How** do the mining results relate to the input graph topology?
- **Why** are two seemingly different users in the same community?
- **Why** is a particular tweet more likely to go viral than another?
- **Why** does the algorithm ‘think’ a transaction looks suspicious?

Algorithmic Fairness in Machine Learning

- **Motivation**
  - No data and/or model are perfect
  - Model trained on data could systematically harm a group of people
- **Goals:** (1) understand and (2) correct the bias(es)
- **Examples:** bias in machine learning systems


REPORT \ TECH \ ARTIFICIAL INTELLIGENCE

What a machine learning tool that turns Obama white can (and can't) tell us about AI bias

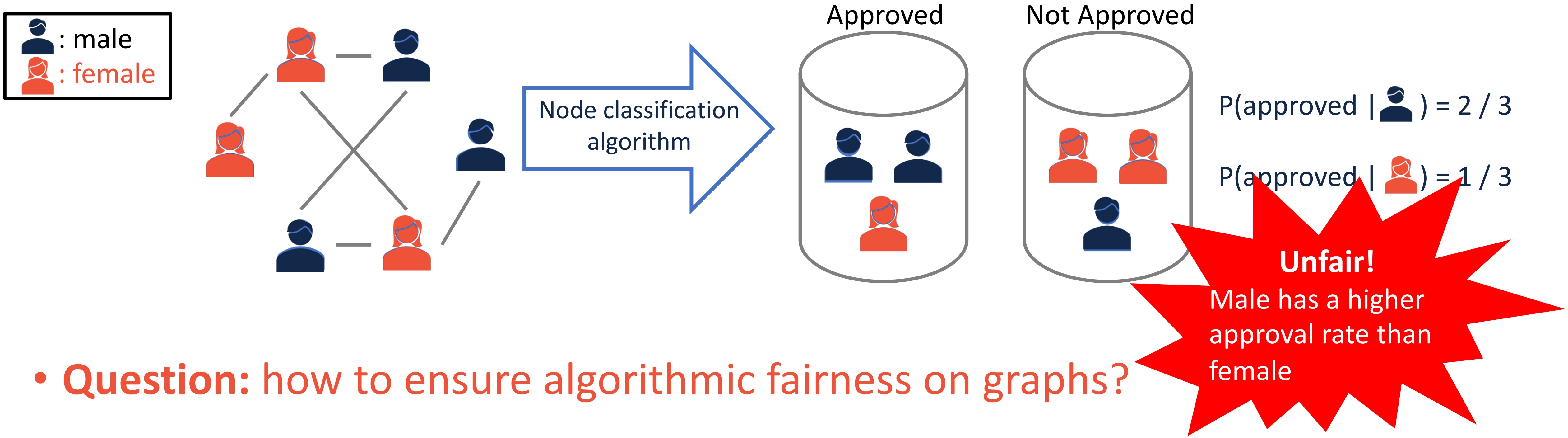
A striking image that only hints at a much bigger problem

By James Vincent | Jun 23, 2020, 3:45pm EDT

[1] https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-pulse-stylegan-obama-bias

Algorithmic Fairness on Graphs

- **Example:** loan approval

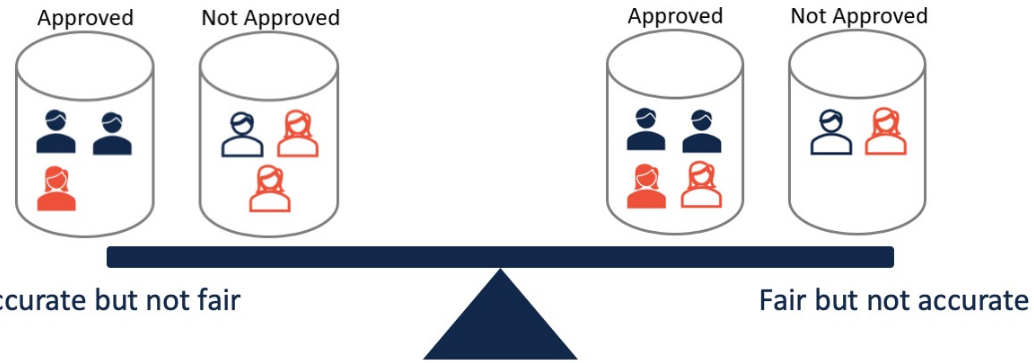


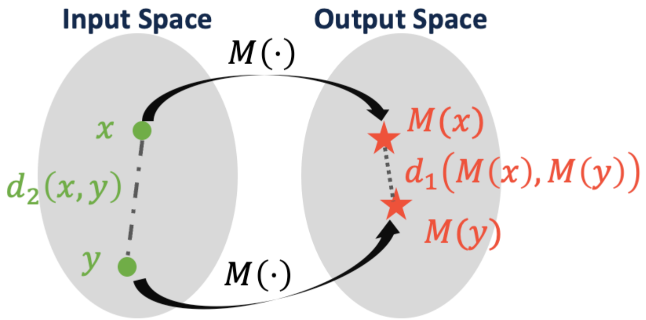
- **Question:** how to ensure algorithmic fairness on graphs?

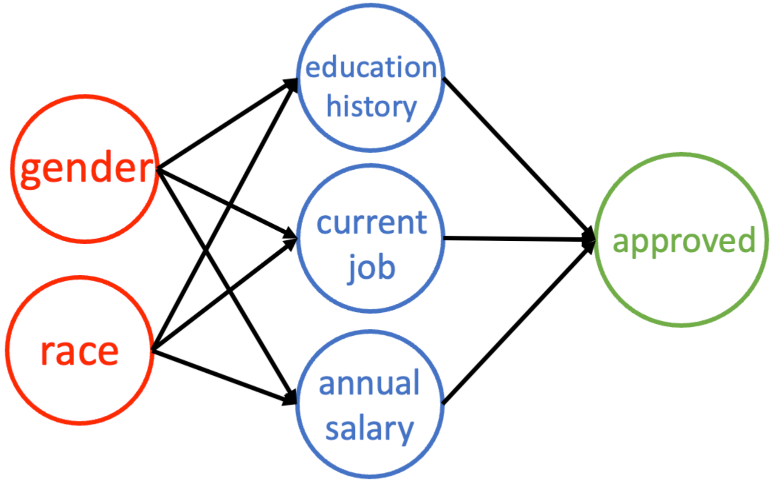
[1] http://tonghanghang.org/netfair.html
[2] http://jiank2.web.illinois.edu/tutorial/cikm21/fair_graph_mining.html

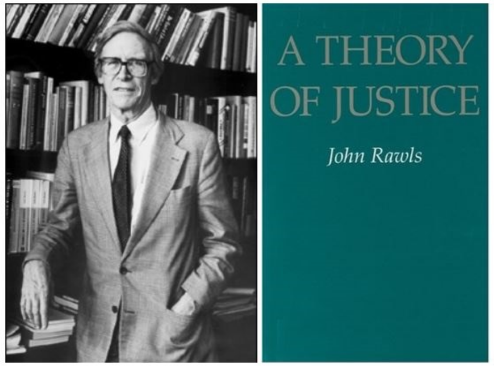
Algorithmic Fairness: Definition

- **Principle:** lack of favoritism from one side or another
- **Definitions of algorithmic fairness**
  - Group fairness
    - Statistical parity
    - Equal opportunity
    - Equalized odds
    - …
  - Individual fairness
  - Counterfactual fairness
  - Difference principle
  - …









[1] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S.. Certifying and Removing Disparate Impact. KDD 2015.
[2] Hardt, M., Price, E., & Srebro, N.. Equality of Opportunity in Supervised Learning. NeurIPS 2016.
[3] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R.. Fairness through Awareness. ITCS 2012.
[4] Kusner, M. J., Loftus, J., Russell, C., & Silva, R.. Counterfactual Fairness. NeurIPS 2017.
[5] Rawls, J.. A Theory of Justice. Press, Cambridge 1971.

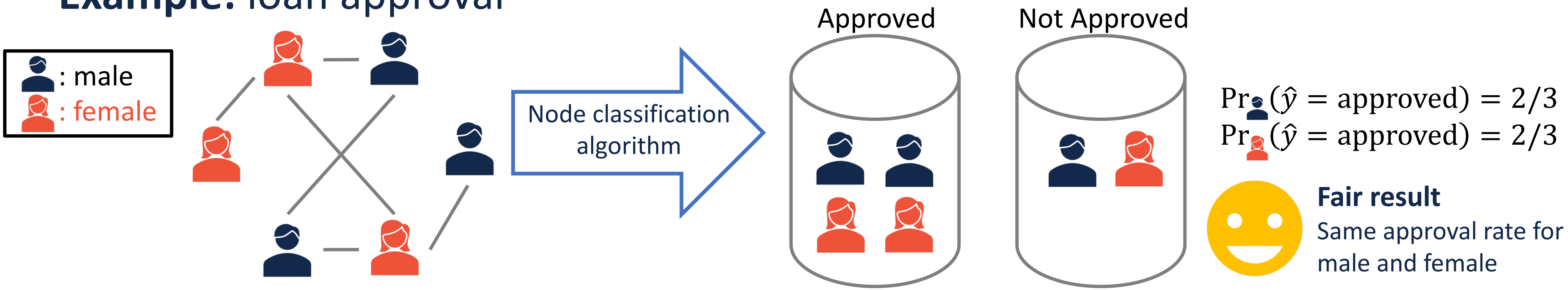
Group Fairness: Statistical Parity

- **Definition:** equal acceptance rate

$$\mathrm{Pr}_{+}(\hat{y} = c) = \mathrm{Pr}_{-}(\hat{y} = c)$$

  – $\hat{y}$: model prediction
  – Pr_{+}: probability for the protected group
  – Pr_{-}: probability for the unprotected group
  – Also known as demographic parity, disparate impact

- **Example:** loan approval



[1] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S.. Certifying and Removing Disparate Impact. KDD 2015.

Group Fairness: Equal Opportunity

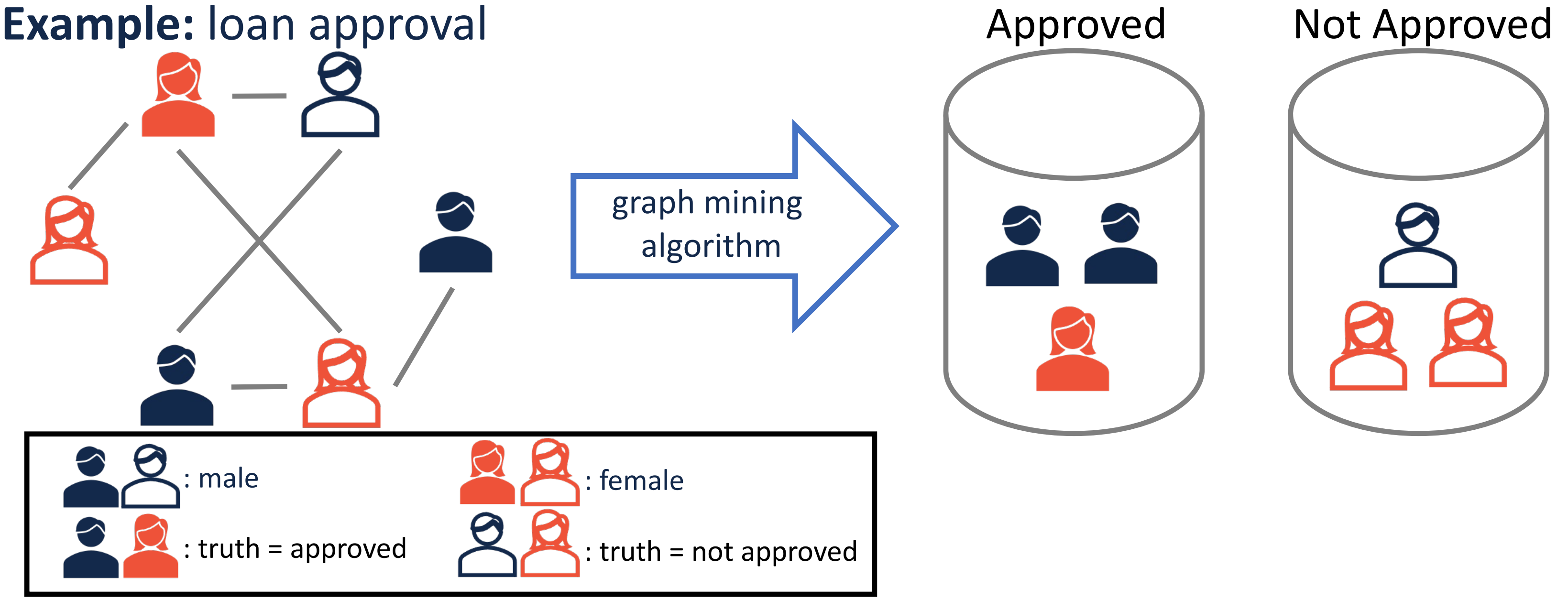
- **Definition:** equal true positive rate

$$\boxed{\Pr_{+}(\hat{y} = c|y = c) = \Pr_{-}(\hat{y} = c|y = c)}$$

If hold for all classes, it is called **equalized odds**

  - y: true label
  - $\hat{y}$: model prediction
  - $\Pr_{+}$: probability for the protected group
  - $\Pr_{-}$: probability for the unprotected group

- **Example:** loan approval



$\Pr(\hat{y} = \text{approved}|) = 1$
$\Pr(\hat{y} = \text{approved}|) = 1$

Fair result
Same true positive rate for male and female

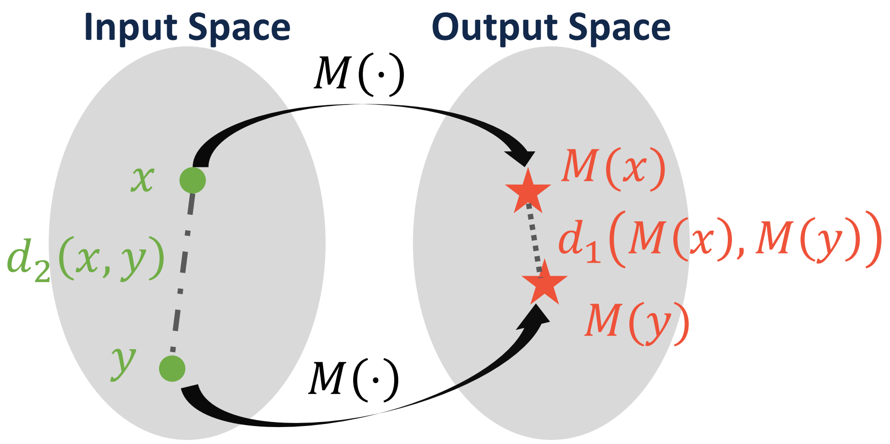
[1] Hardt, M., Price, E., & Srebro, N.. Equality of Opportunity in Supervised Learning. NeurIPS 2016.

Individual Fairness

- **Definition:** similar individuals should have similar outcomes
- **Formulation:** Lipschitz inequality (most common)

$$d_1\big(M(x), M(y)\big) \le L d_2(x, y)$$

  - M: a mapping from input to output
  - d_1: distance metric for output
  - d_2: distance metric for input
  - L: a constant scalar
- **Example**



[1] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R.. Fairness through Awareness. ITCS 2012.

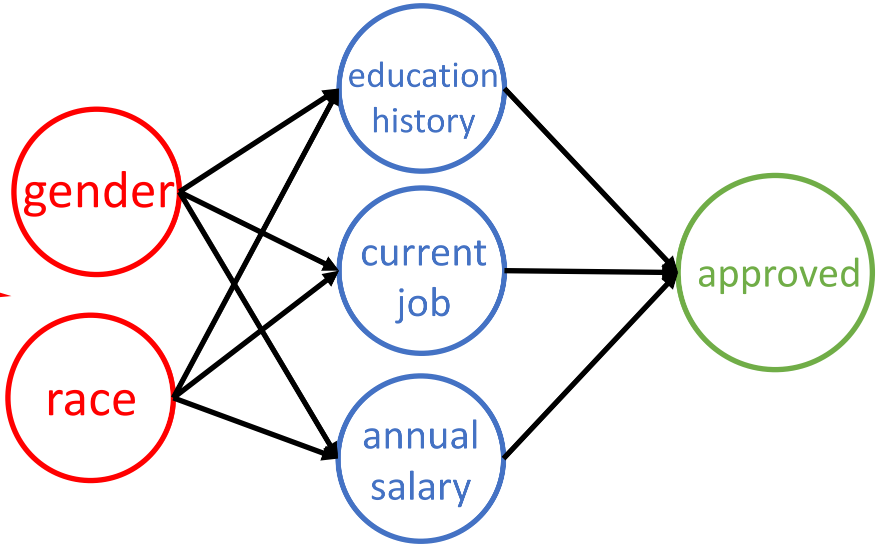
Counterfactual Fairness

counterfactual version

- **Definition:** same outcomes for ‘different versions’ of the same candidate

$$\Pr\left(\hat{y}_{s=s_1} = c \mid s = s_1, x = \mathbf{x}\right) = \Pr\left(\hat{y}_{s=s_2} = c \mid s = s_2, x = \mathbf{x}\right)$$

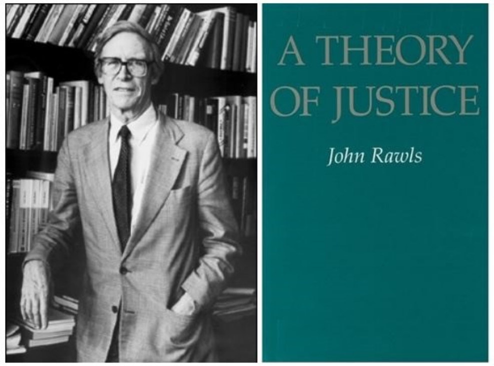
  - $\Pr\left(\hat{y}_{s=s_1} = c \mid s = s_1, x = \mathbf{x}\right)$: version 1 of $\mathbf{x}$ with sensitive demographic s_1
  - $\Pr\left(\hat{y}_{s=s_2} = c \mid s = s_2, x = \mathbf{x}\right)$: version 2 of $\mathbf{x}$ with sensitive demographic s_2
- **Example:** causal graph of loan approval



[1] Kusner, M. J., Loftus, J., Russell, C., & Silva, R.. Counterfactual Fairness. NeurIPS 2017.

Rawlsian Difference Principle

- **Origin:** distributive justice
- **Goal:** fairness as just allocation of social welfare



"Inequalities are permissible when they maximize […] the long-term expectations of the least fortunate group."

-- John Rawls, 1971

- **Formulation:** max-min problem
  – **Min:** the least fortunate group with smallest welfare/utility
  – **Max:** maximization of the corresponding utility
- Also known as max-min fairness, accuracy disparity

[1] Rawls, J.. A Theory of Justice. Press, Cambridge 1971.

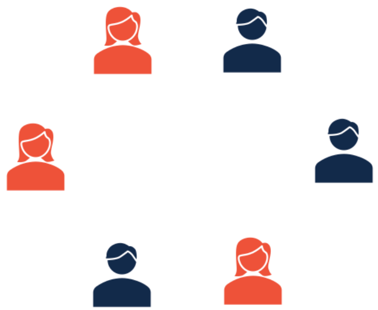
- **Justice as fairness**
  - Justice is a virtue of instituitions
  - Free persons enjoy and acknowledge the rules
- **Well-ordered society**
  - Designed to advance the good of its members
  - Regulated by a public conception of justice

Challenge #1: Theoretical Challenge

- **Assumption**

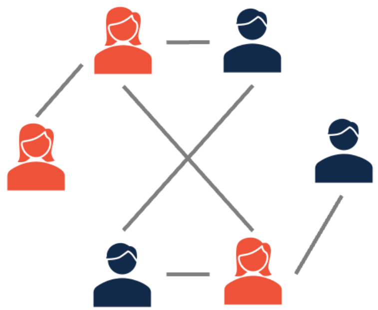
	Classic machine learning	**Graph mining**
Data	IID samples	Non-IID graph

  - **IID:** independent and identically distributed
- **Example**



- Individuals are independent
- Cannot affect others

Classic machine learning



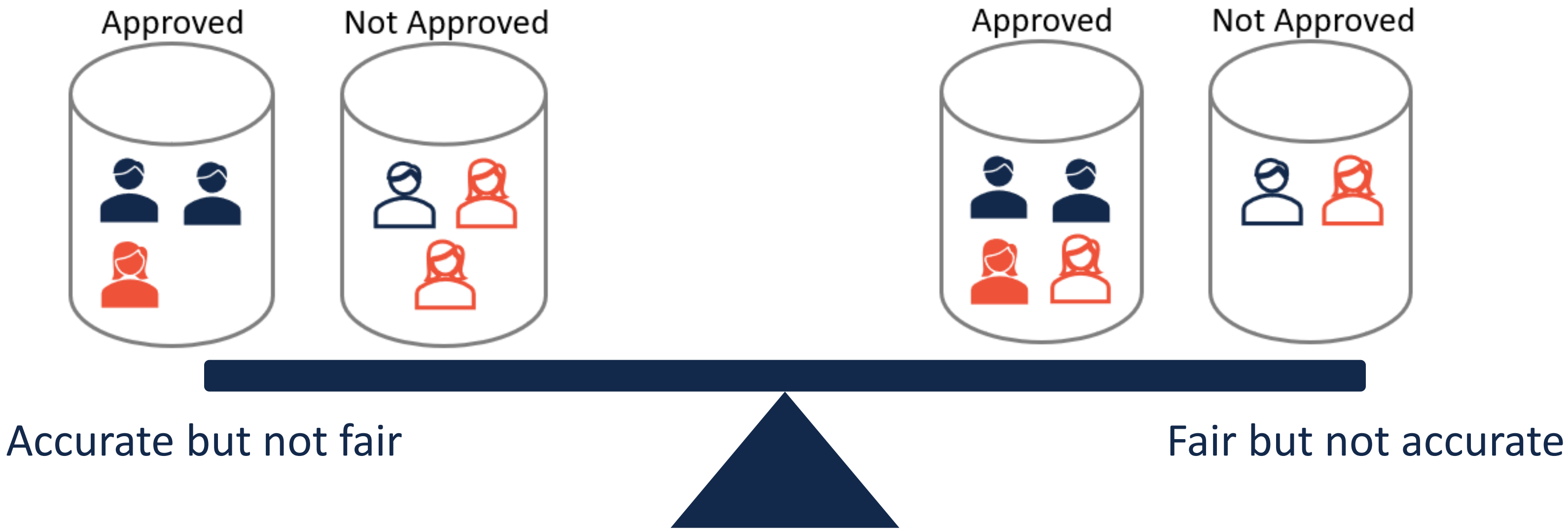
- Individuals are connected
- Can affect others through connection(s)

Graph mining

- **Challenges:** implication of non-IID nature on
  - Measuring bias
    - Dyadic fairness, degree-related fairness
  - Mitigating unfairness
    - Enforce fairness by graph structure imputation

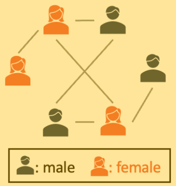
Challenge #2: Algorithmic Challenge

- **Dilemma:** utility vs. fairness
- **Example:** loan approval
  - Utility = classification accuracy
  - Fairness = statistical parity

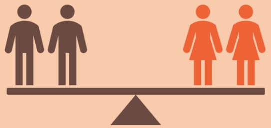


- **Questions**
  - Can we improve fairness at no cost of utility?
  - If not, how to balance the trade-off between utility and fairness?

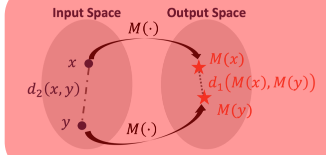
Roadmap



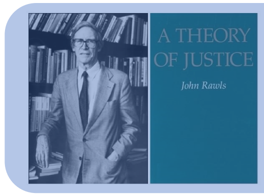
Introduction



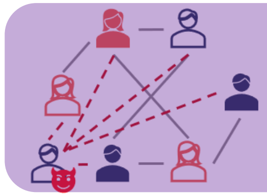
Part I: Group Fairness on Graphs



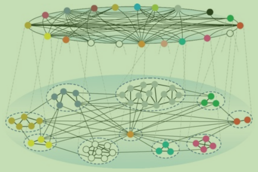
Part II: Individual Fairness on Graphs



Part III: Other Fairness on Graphs



Part IV: Beyond Fairness on Graphs



Part V: Future Trends

Overview of Part I

Group Fairness on Graphs

- **Ranking**
  - Fair PageRank
- **Clustering**
  - Fair Spectral Clustering
- **Embedding**
  - Fairwalk
  - CrossWalk
  - Compositional Fairness
  - FairGNN

Preliminary: PageRank

- **Assumption:** important webpage → linked by many others
- **Formulation**
  - Iterative method for the following linear system

$$\mathbf{r} = c\mathbf{A}^T\mathbf{r} + (1-c)\mathbf{e}$$

    - $\mathbf{A}$: transition matrix
    - $\mathbf{r}$: PageRank vector
    - c: damping factor
    - $\mathbf{e}$: teleportation vector
  - Closed-form solution

$$\mathbf{r} = (1-c)(\mathbf{I} - c\mathbf{A}^T)^{-1}\mathbf{e}$$

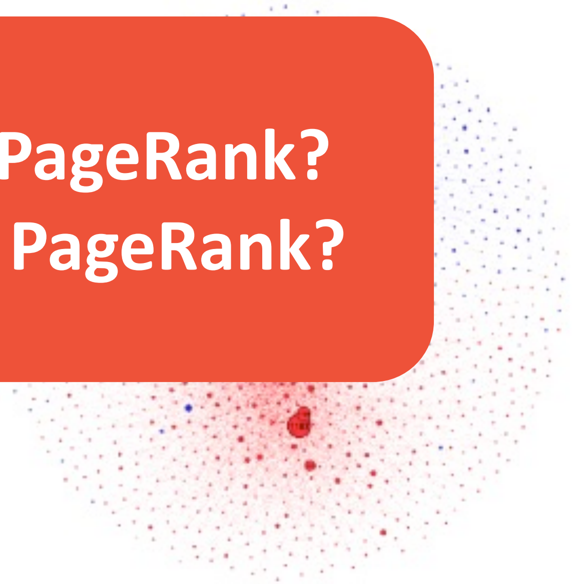
- **Variants**
  - Personalized PageRank (PPR)
  - Random Walk with Restart (RWR)
  - …

[1] Page, L., Brin, S., Motwani, R., & Winograd, T.. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab 1999.
[2] Haveliwala, T. H.. Topic-sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. TKDE 2003.
[3] Tong, H., Faloutsos, C., & Pan, J. Y.. Fast Random Walk with Restart and Its Applications. ICDM 2006.

Unfairness in PageRank

- **PageRank score:** a measure of node importance in the network
- **Facts:** some nodes hold more important/central positions in the network
  - biased academic ranking w.r.t. gender → underestimation of scientific contribution by female
- **Example**
  - **Network:** 1
  - **Groups:** re
  - **Red node**
    - ~48% o
    - ~33% o

1. How to define group fairness for PageRank?
2. Can we enforce group fairness on PageRank?

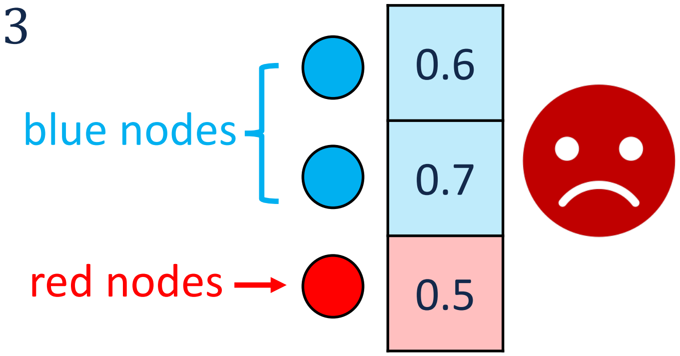


Unfair ranking
Similar number of red nodes vs. blue nodes (48% red vs. 52% blue)
Much less PageRank mass of red nodes (33% red vs. 67% blue)

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.
[2] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

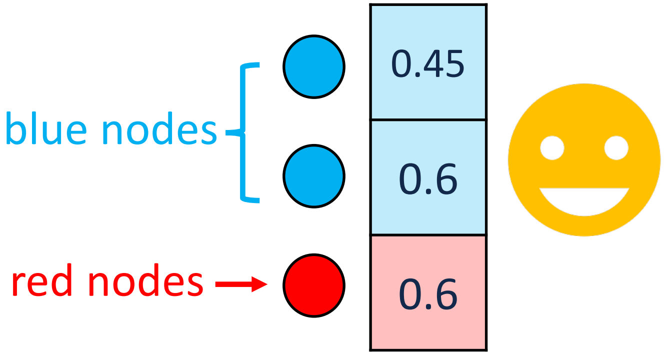
Fairness Measure: ϕ-Fairness

- **Given:** (1) a graph G; (2) a parameter ϕ
- **Definition:** a PageRank vector is ϕ-fair if at least ϕ fraction of total PageRank mass is allocated to the protected group
- **Variants and generalizations**
  - Statistical parity → ϕ = fraction of protected group
  - Affirmative action → ϕ = a desired ratio (e.g., 20%)
- **Example**
  - Protected group = red nodes
  - $\phi = 1/3$



Not ϕ-fair!

$$\frac{0.5}{0.6 + 0.7 + 0.5} < \frac{1}{3}$$



ϕ-fair!

$$\frac{0.6}{0.45 + 0.6 + 0.6} \geq \frac{1}{3}$$

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.
[2] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

Problem Definition: Fair PageRank

- **Given**
  - A graph with transition matrix $\mathbf{A}$
  - Partitions of nodes
    - Red nodes ($\mathcal{R}$): protected group
    - Blue nodes ($\mathcal{B}$): unprotected group
- **Find:** a fair PageRank vector $\tilde{\mathbf{r}}$ that is
  - ϕ-fair
  - Close to the original PageRank vector $\mathbf{r}$

Fair PageRank: Solutions

- **Recap:** closed-form solution for PageRank

$$\mathbf{r} = (1 - c)(\mathbf{I} - c\mathbf{A}^T)^{-1}\mathbf{e}$$

- **Parameters in PageRank**
  - **Damping factor** c avoids sinks in the random walk (i.e., nodes without outgoing links)
  - **Teleportation vector e** controls the starting node where a random walker restarts
    - Can we control where the walker teleports to? ⟵ Solution #1: fairness-sensitive PageRank
  - **Transition matrix** $\mathbf{A}$ controls the next step where the walker goes to
    - Can we modify the transition probabilities?
    - Can we modify the graph structure?

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Solution #1: Fairness-sensitive PageRank

- **Intuition**
  - Find a teleportation vector $\mathbf{e}$ to make PageRank vector ϕ-fair
  - Keep transition matrix $\mathbf{A}$ and $\mathbf{Q}^T = (1-c)(\mathbf{I} - c\mathbf{A}^T)^{-\mathbf{1}}$ fixed
- **Observation:** mass of PageRank $\mathbf{r}$ w.r.t. red nodes $\mathcal{R}$

$$\mathbf{r}(\mathcal{R}) = \mathbf{Q}^T[\mathcal{R}, :]\mathbf{e}$$

  - $\mathbf{Q}^T[\mathcal{R}, :]$: rows of $\mathbf{Q}^T$ w.r.t. nodes in set $\mathcal{R}$
- **(Convex) optimization problem**

$$\begin{aligned} \min_{\mathbf{e}} \quad & \boxed{\|\mathbf{Q}^T\mathbf{e} - \mathbf{r}\|^2} \\ \text{s.t.} \quad & \boxed{\begin{array}{l} \mathbf{e}[i] \in [0, 1], \forall i \\ \|\mathbf{e}\|_1 = 1 \end{array}} \\ & \boxed{\|\mathbf{Q}^T[\mathcal{R}, :]\mathbf{e}\|_1 = \phi} \end{aligned}$$

The fair PageRank $\mathbf{Q}^T\mathbf{e}$ is as close as possible to the original PageRank $\mathbf{r}$

The teleportation vector $\mathbf{e}$ is a probability distribution

The fair PageRank $\mathbf{Q}^T\mathbf{e}$ needs to be ϕ-fair

  - Can be solved by any convex optimization solvers

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Fairness-sensitive PageRank: Example

- **Settings:** $\phi = 1/3$ and protected node = red node
- **Original PageRank**

$$
\begin{array}{l} \text{rows w.r.t. blue nodes} \\ \\ \text{row w.r.t. red nodes} \end{array}
\overset{\mathbf{Q}^T}{\begin{bmatrix} 0.8 & 0.7 & 0.3 \\ 0.7 & 0.9 & 0.5 \\ 0.3 & 0.5 & 0.7 \end{bmatrix}}
\quad
\overset{\mathbf{e}}{\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}}
\qquad
\mathbf{r} = \mathbf{Q}^T\mathbf{e} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.5 \end{bmatrix}
$$

Not ϕ-fair!

$$\frac{0.5}{0.6 + 0.7 + 0.5} < \frac{1}{3}$$

- **Fairness-sensitive PageRank**

$$
\begin{array}{l} \text{rows w.r.t. blue nodes} \\ \\ \text{row w.r.t. red nodes} \end{array}
\overset{\mathbf{Q}^T}{\begin{bmatrix} 0.8 & 0.7 & 0.3 \\ 0.7 & 0.9 & 0.5 \\ 0.3 & 0.5 & 0.7 \end{bmatrix}}
\quad
\overset{\tilde{\mathbf{e}}}{\begin{bmatrix} 1/6 \\ 1/6 \\ 2/3 \end{bmatrix}}
\qquad
\tilde{\mathbf{r}} = \mathbf{Q}^T\tilde{\mathbf{e}} = \begin{bmatrix} 0.45 \\ 0.6 \\ 0.6 \end{bmatrix}
$$

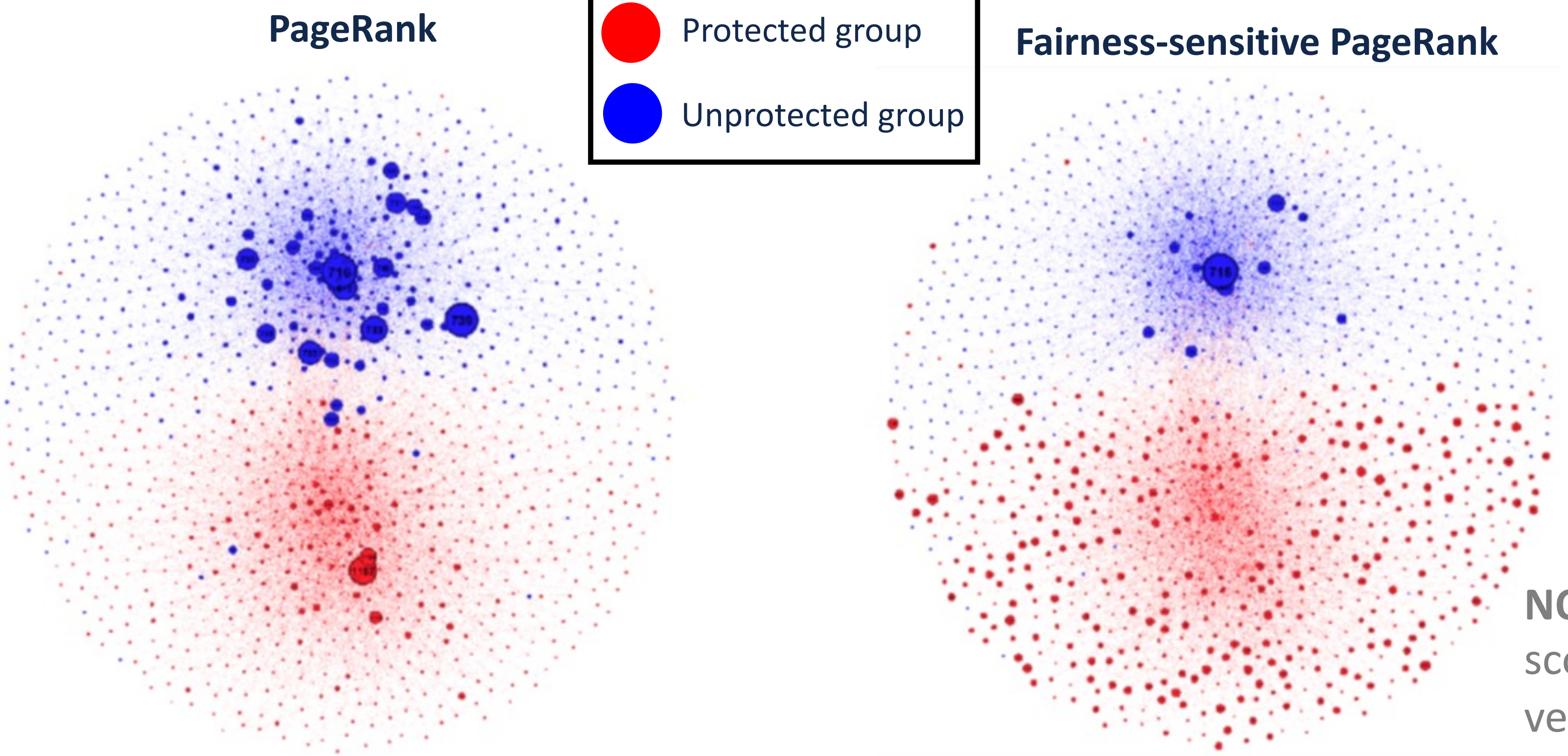
ϕ-fair!

$$\frac{0.6}{0.45 + 0.6 + 0.6} \geq \frac{1}{3}$$

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Fairness-sensitive PageRank: Experiment

- **Observation:** the teleportation vector allocates more weight to the red nodes, especially nodes at the periphery of the network
  - – More likely to (1) restart at red nodes and (2) walk to other red nodes more often



NOTE: size is proportional to score in the teleportation vector

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Fair PageRank: Solutions

- **Recap:** closed-form solution for PageRank

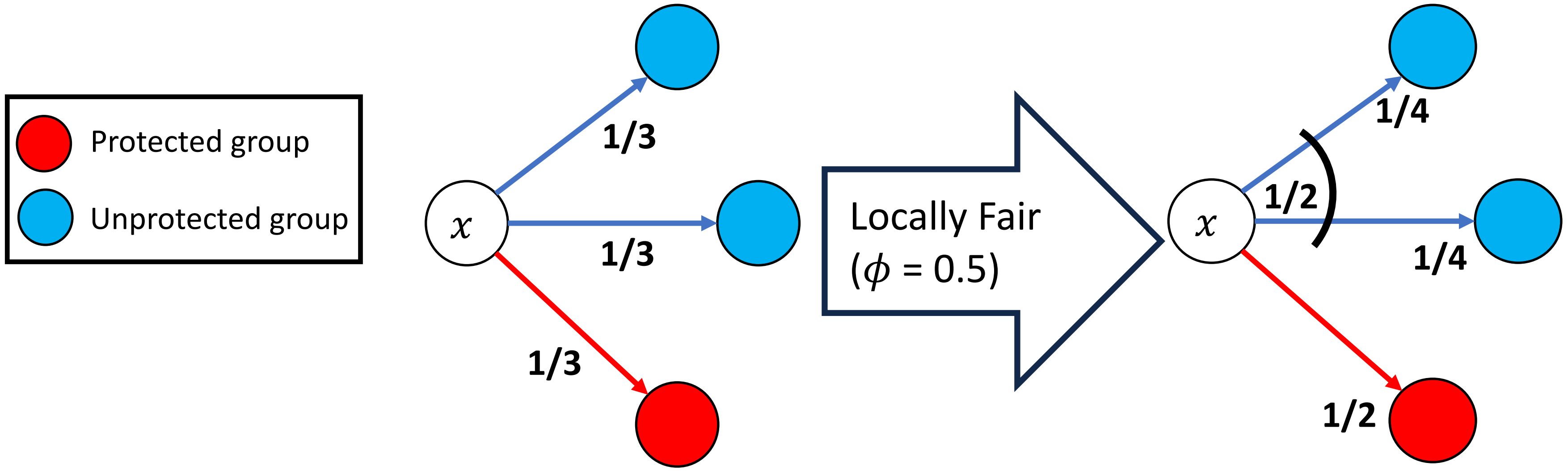
$$\mathbf{r} = (1 - c)(\mathbf{I} - c\mathbf{A}^T)^{-1}\mathbf{e}$$

- **Parameters in PageRank**
  - **Damping factor** c avoids sinks in the random walk (i.e., nodes without outgoing links)
  - **Teleportation vector e** controls the starting node where a random walker restarts
    - Can we control where the walker teleports to?
  - **Transition matrix** $\mathbf{A}$ controls the next step where the walker goes to
    - Can we modify the transition probabilities? ← Solution #2: locally fair PageRank
    - Can we modify the graph structure?

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Solution #2: Locally Fair PageRank

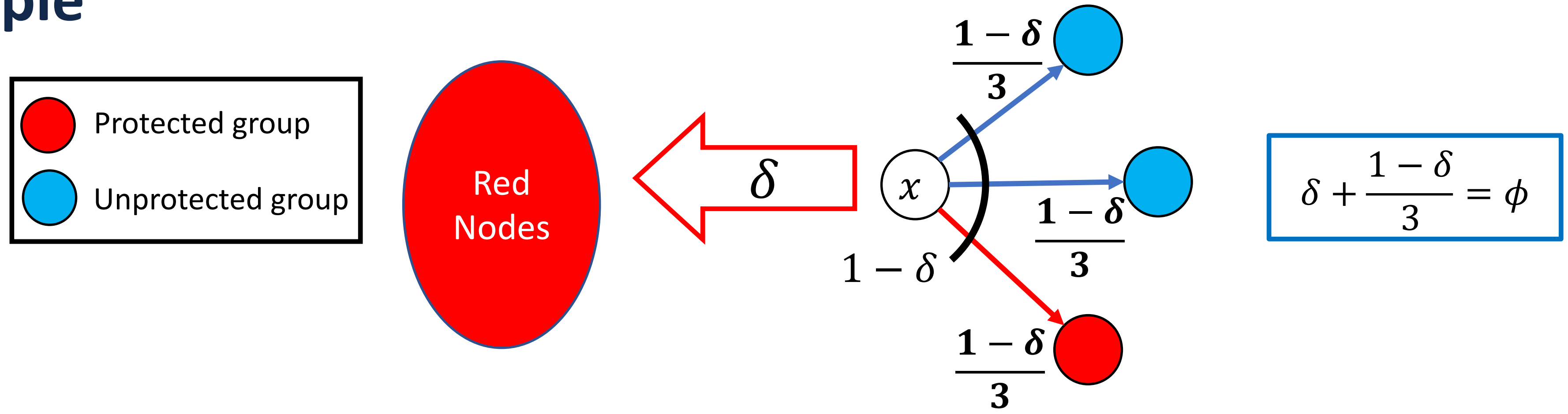
- **Intuition:** adjust the transition matrix A to obtain a fair random walk
- **Neighborhood locally fair PageRank**
  - **Key idea:** jump with probability ϕ to red nodes and (1- ϕ) to blue nodes
  - **Example**



[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Solution #2: Locally Fair PageRank

- **Residual locally fair PageRank**
  - **Key idea:** jump with
    - Equal probability to 1-hop neighbors
    - A residual probability δ to the other red nodes
  - **Example**



- **Residual allocation policies:** neighborhood allocation, uniform allocation, proportional allocation, optimized allocation

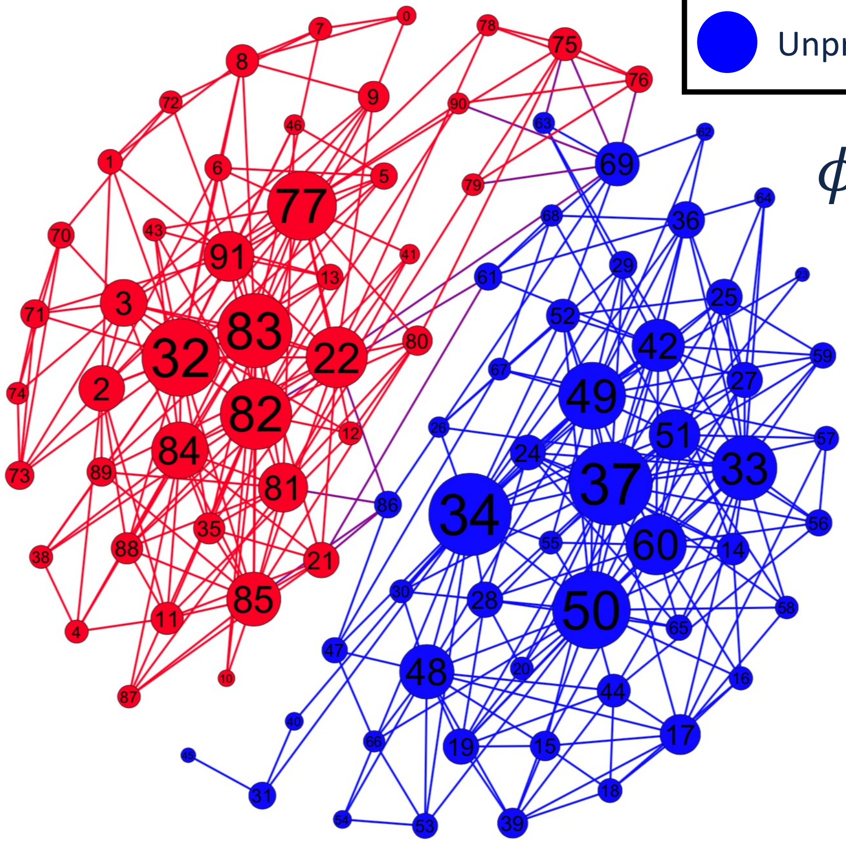
[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

- **Neighborhood allocation:** allocate the residual to protected neighbors, equivalent to neighborhood locally fair PageRank
- **Uniform allocation:** uniformly allocate the residual to all protected nodes
- **Proportional allocation:** allocated the residual to all protected nodes proportionally to their PageRank score
- **Optimized allocation:** allocate the residual to all protected nodes while minimizing the difference with original PageRank score

Locally Fair PageRank: Experiment

- **Observation:** PageRank weight is shifted to the blue nodes at boundary

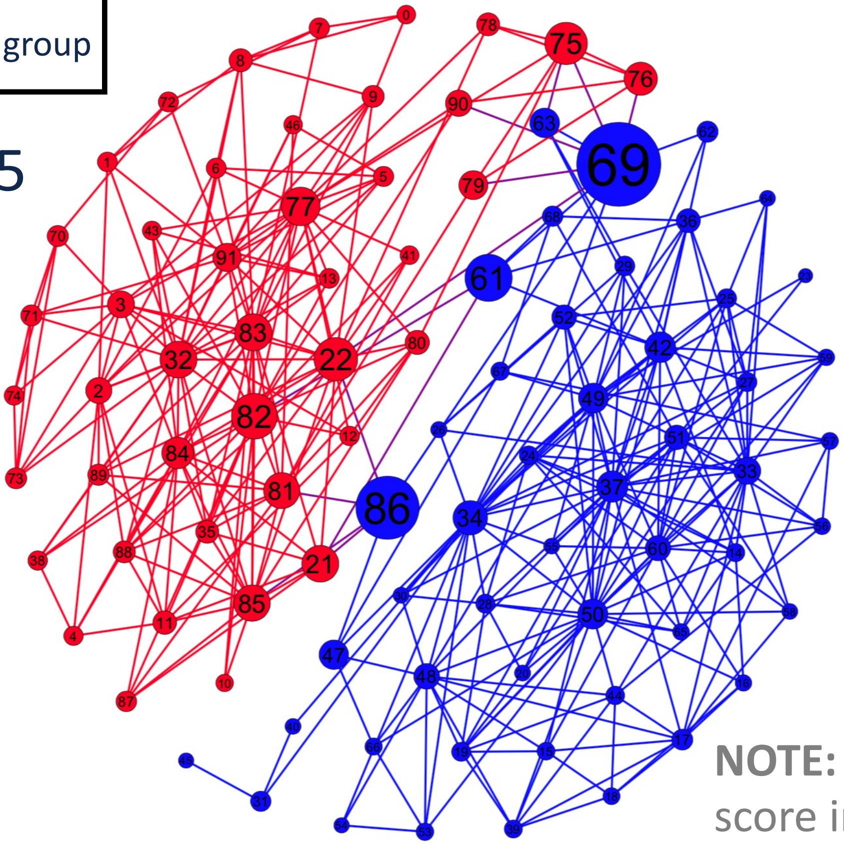
PageRank



Protected group
Unprotected group

$\phi = 0.5$

Neighborhood Locally Fair PageRank



NOTE: size is proportional to score in the PageRank vector

[1] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

Fair PageRank: Solutions

- **Recap:** closed-form solution for PageRank

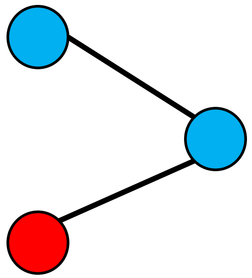
$$\mathbf{r} = (1 - c)(\mathbf{I} - c\mathbf{A}^T)^{-1}\mathbf{e}$$

- **Parameters in PageRank**
  - **Damping factor** c avoids sinks in the random walk (i.e., nodes without outgoing links)
  - **Teleportation vector** $\mathbf{e}$ controls the starting node where a random walker restarts
    - Can we control where the walker teleports to?
  - **Transition matrix** $\mathbf{A}$ controls the next step where the walker goes to
    - Can we modify the transition probabilities?
    - Can we modify the graph structure? ← Solution #3: best fair edge identification

[1] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

Solution #3: Best Fair Edge Identification

- **Intuition:** add edges that can improve the PageRank fairness to the graph
- **Example**
  - = protected node
  - $\phi = 1/3$

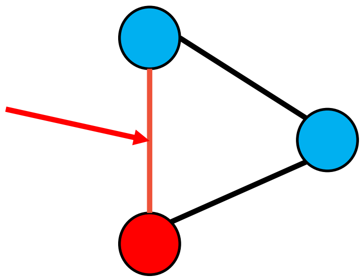


$$\mathbf{r} = \mathbf{Q}^T\mathbf{e} = \begin{bmatrix} 0.257 \\ 0.486 \\ 0.257 \end{bmatrix}$$

Not ϕ-fair!

$$\frac{0.257}{0.257 + 0.486 + 0.257} < \frac{1}{3}$$

New edge to add



$$\tilde{\mathbf{r}} = \tilde{\mathbf{Q}}^T\mathbf{e} = \begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix}$$

ϕ-fair!

$$\frac{0.333}{0.333 + 0.333 + 0.333} = \frac{1}{3}$$

- **Question:** how to find the edges with the highest improvement?

[1] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

Best Fair Edge Identification: Problem Definition

- **Given**
  - $G = (\mathcal{V}, \mathcal{E})$
    - $\mathcal{E}$: edge set
    - $\mathcal{V}$: node set
  - $\mathcal{S} \subseteq \mathcal{V}$: protected node set
  - $p_{\mathcal{E}}(\mathcal{S}) = \sum_{i \in \mathcal{V}} p_{\mathcal{E}}(i)$: total PageRank mass of nodes in $\mathcal{S}$ on graph with edge set $\mathcal{E}$
- **Fairness gain of edge addition**

$$\text{gain}(x, y) = \boxed{p_{\mathcal{E} \cup (x,y)}(\mathcal{S}) - p_{\mathcal{E}}(\mathcal{S})}$$

Naive method
Exhaustively recompute PageRank with the addition of **each** node pair

- **Goal:** find the edge $(x, y), \forall x, y \in \mathcal{V}$, such that

$$\underset{(x,y)}{\text{argmax}} \quad \text{gain}(x, y)$$

- **Question:** how to efficiently compute the gain?

[1] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

Best Fair Edge Identification: Fairness Gain

- **Main result:** for a node x, the gain of adding a link to another node y

$$\text{gain}(x,y) = \Lambda(x,y) p_{\mathcal{E}}(x)$$

where $\Lambda(x,y)$ has the form

The 'sensitivity' of target node y

The average 'sensitivity' of source node x's neighbors

$$\Lambda(x,y) = \frac{\frac{c}{1-c}\left(p_{\mathcal{E}}(\mathcal{S}|y) - \frac{1}{d_x}\sum_{u\in\mathcal{N}_x} p_{\mathcal{E}}(\mathcal{S}|u)\right)}{d_x + \frac{c}{1-c}\left(\frac{1}{d_x}\sum_{u\in\mathcal{N}_x} p_{\mathcal{E}}(x|u) - p_{\mathcal{E}}(x|y)\right) + 1}$$

degree of source node

Average proximity of node x's neighbors to x

  - $p_{\mathcal{E}}(x|y)$: personalized PageRank (PPR) score of node x, with query node y, based on edge set $\mathcal{E}$
  - $p_{\mathcal{E}}(\mathcal{S}|y) = \sum_{i\in\mathcal{S}} p_{\mathcal{E}}(i|y)$: total PPR mass of nodes in $\mathcal{S}$, with query node y, based on edge set $\mathcal{E}$
- $p_{\mathcal{E}}(x)$: node x should have high PageRank score
- d_x: node x should have small degree
- $p_{\mathcal{E}}(x|y) - \frac{1}{d_x}\sum_{u\in\mathcal{N}_x} p_{\mathcal{E}}(x|u)$: node y is close to node x
- $p_{\mathcal{E}}(\mathcal{S}|y) - \frac{1}{d_x}\sum_{u\in\mathcal{N}_x} p_{\mathcal{E}}(\mathcal{S}|u)$: node y is more sensitive than the source node x's neighborhood

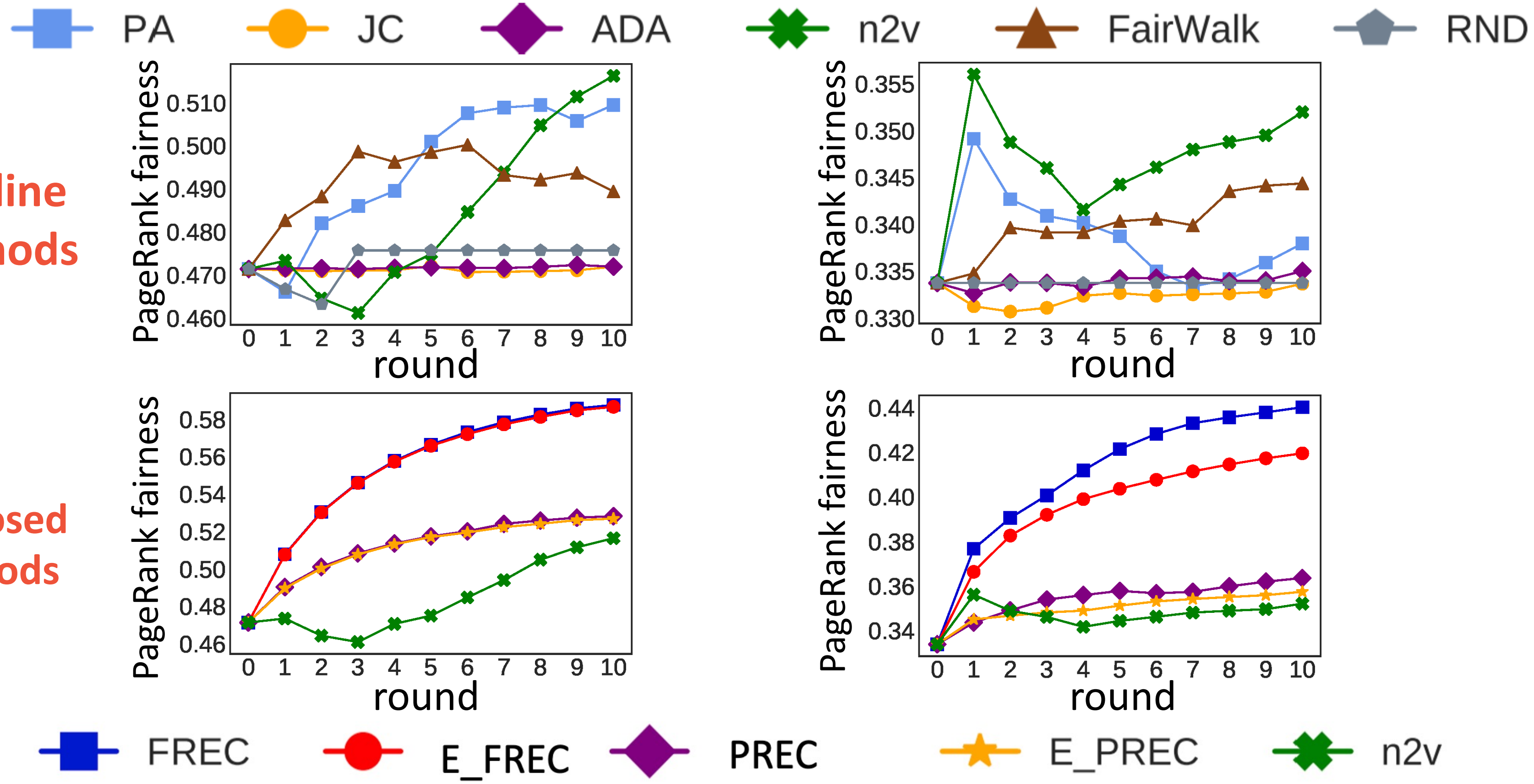
[1] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

Best Fair Edge Identification: Experiment

- **Observation:** the proposed method find the best edges to improve PageRank fairness

Baseline Methods

Proposed Methods



Books Dataset **Blogs Dataset**

[1] Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.

- FREC: select edge (x, y) with highest $\mathrm{gain}(x, y) = \Lambda(x, y) p_{\varepsilon}(x)$
- PREC: select edge (x, y) with highest $\mathrm{gain}(x, y \mid x) = \Lambda(x, y) p_{\varepsilon}(x|x)$
- E_FREC: select edge (x, y) with highest $\mathrm{gain}(x, y) p_{\mathrm{acc}}(x, y)$
- E_PREC: select edge (x, y) with highest $\mathrm{gain}(x, y \mid x) p_{\mathrm{acc}}(x, y)$

* $p_{\mathrm{acc}}(x, y)$: prediction probability by a logistic regression classifier on the existence of (x, y) using node2vec embeddings

Overview of Part I

- Group Fairness on Graphs
  - Ranking
    - Fair PageRank
  - Clustering
    - Fair Spectral Clustering
  - Embedding
    - Fairwalk
    - CrossWalk
    - Compositional Fairness
    - FairGNN

Preliminary: Spectral Clustering (SC)

- **Goal:** find k clusters such that
  - maximize intra-connectivity
  - minimize inter-connectivity
- **Optimization problem**

Ratio cut

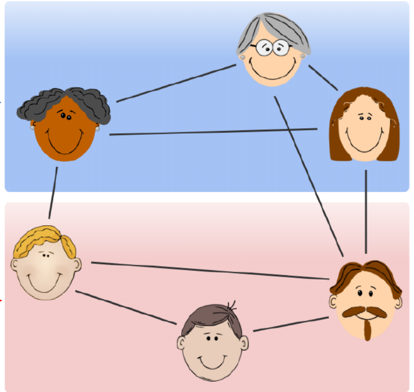
$$\min_{\mathbf{U}} \ \boxed{\mathrm{Tr}\left(\mathbf{U}^T\mathbf{L}\mathbf{U}\right)} \qquad \text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

where $\mathbf{L}$ is Laplacian matrix of $\mathbf{A}$, $\mathbf{U}$ is a matrix with k orthonormal column vectors

- **Solution:** rank-k eigen-decomposition
  - $\mathbf{U}$ = eigenvectors with k smallest eigenvalues
- **Example**

All female individuals are grouped together

All male individuals are grouped together



Unfair clustering
The clustering results are highly correlated with gender

[1] Ng, A. Y., Jordan, M. I., & Weiss, Y.. On Spectral Clustering: Analysis and an Algorithm. NeurIPS 2002.
[2] Shi, J., & Malik, J.. Normalized Cuts and Image Segmentation. TPAMI 2000.

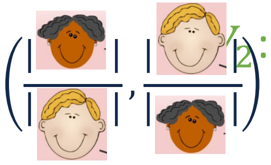
Fairness Measure: Balance Score

- **Intuition:** fairness as balance among clusters
- **Given:** a node set V with
  - h demographic groups: $V = V_1 \cup V_2 \dots \cup V_h$
  - k clusters: $V = C_1 \cup C_2 \dots \cup C_k$
- **Definition**

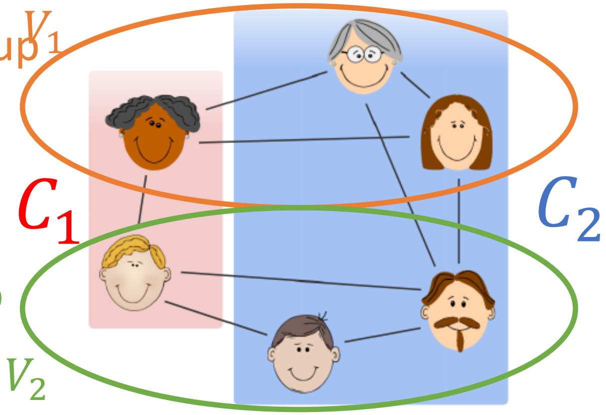
$$\text{balance}(C_l) = \min_{s \neq s' \in [h]} \frac{|V_s \cap C_l|}{|V_{s'} \cap C_l|} \in [0, 1], \qquad \forall l \in [1,2, \dots, k]$$

- **Intuition:** higher balance → fairer
  - Each demographic group is presented with similar fractions as in the whole dataset for every cluster
- **Example**

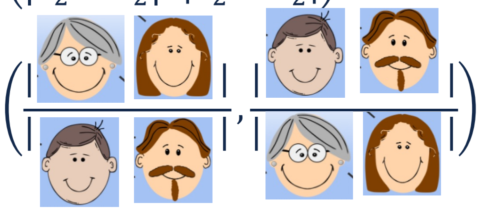
$$\text{balance}(C_1) = \min\left(\frac{|V_1 \cap C_1|}{|V_2 \cap C_1|}, \frac{|V_2 \cap C_1|}{|V_1 \cap C_1|}\right)$$

$= \min$ 

$= 1$



$$\text{balance}(C_2) = \min\left(\frac{|V_1 \cap C_2|}{|V_2 \cap C_2|}, \frac{|V_2 \cap C_2|}{|V_2 \cap C_2|}\right)$$

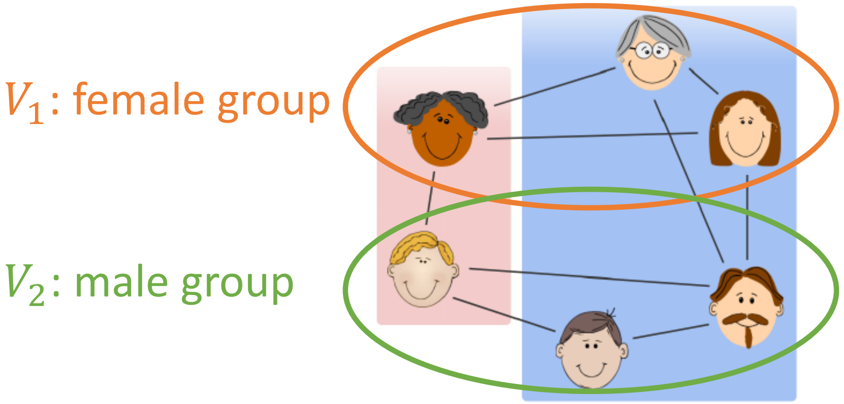
$= \min$ 

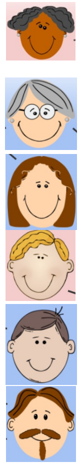
$= 1$

[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.

Fair Spectral Clustering: Formulation

- **Key idea:** fairness as linear constraint
  - **Given**
    - The spectral embedding $\mathbf{U}$ of n nodes in l clusters $(C_1, \dots, C_l)$
    - h demographic groups $(V_1, \dots, V_s)$
  - **Define**
    - $\mathbf{f}^{(s)}[i] = 1$ if $i \in V_s$ and 0 otherwise
    - $\mathbf{F} =$ a matrix with $\mathbf{f}^{(s)} - \left(\frac{|V_s|}{n}\right)\mathbf{1}_n$ $(s \in [1, \dots, h-1])$ as column vectors
  - **Observation:** $\mathbf{F}^T\mathbf{U} = \mathbf{0} \Leftrightarrow$ balanced clusters (i.e., fair clusters)
- **Example**





$\mathbf{f}^{(1)}$	$\mathbf{f}^{(2)}$
1	0
1	0
1	0
0	1
0	1
0	1

Fair fraction

0.5	0.5
0.5	0.5
0.5	0.5
0.5	0.5
0.5	0.5
0.5	0.5

$\mathbf{F} =$

0.5	-0.5
0.5	-0.5
0.5	-0.5
-0.5	0.5
-0.5	0.5
-0.5	0.5

[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.

Fair Spectral Clustering: Solution

- **Optimization problem**

$$\min_{\mathbf{U}} \quad \mathrm{Tr}\,(\mathbf{U}^T\mathbf{L}\mathbf{U}) \qquad \mathrm{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \boxed{\mathbf{F}^T\mathbf{U} = \mathbf{0}}$$

How to solve?

- **Solution**
  - **Observation:** $\mathbf{F}^T\mathbf{U} = \mathbf{0}$ → $\mathbf{U}$ is in the null space of $\mathbf{F}^T$
  - **Steps**
    - Define $\mathbf{Z}$ = orthonormal basis of null space of $\mathbf{F}^T$
    - Rewrite $\mathbf{U} = \mathbf{Z}\mathbf{Y}$

$$\min_{\mathbf{U}} \quad \mathrm{Tr}\,(\mathbf{Y}^T\mathbf{Z}^T\mathbf{L}\mathbf{Z}\mathbf{Y}) \qquad \mathrm{s.t.} \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I}$$

  - **Method:** rank-k eigen-decomposition on $\mathbf{Z}^T\mathbf{L}\mathbf{Z}$

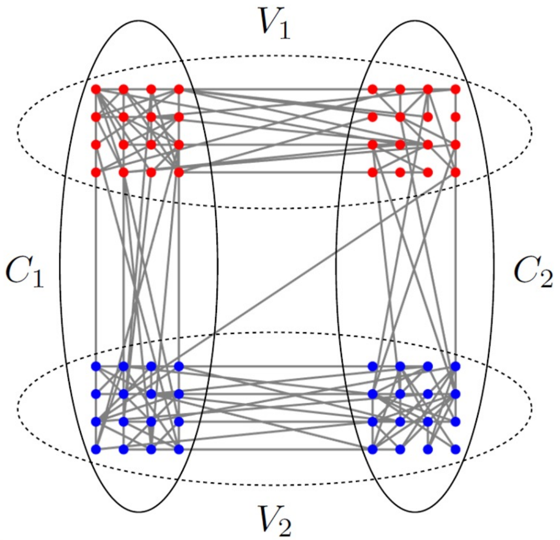
[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.

Fair Spectral Clustering: Correctness

- **Given**
  - A random graph with nodes V by a variant of the Stochastic Block Model (SBM)
  - Edge probability between two nodes i and j

$$\mathrm{P}(i,j) = \begin{cases} a, & i \text{ and } j \text{ in same cluster and in same group} \\ b, & i \text{ and } j \text{ not in same cluster but in same group} \\ c, & i \text{ and } j \text{ in same cluster but not in same group} \\ d, & i \text{ and } j \text{ not in same cluster and not in same group} \end{cases}$$

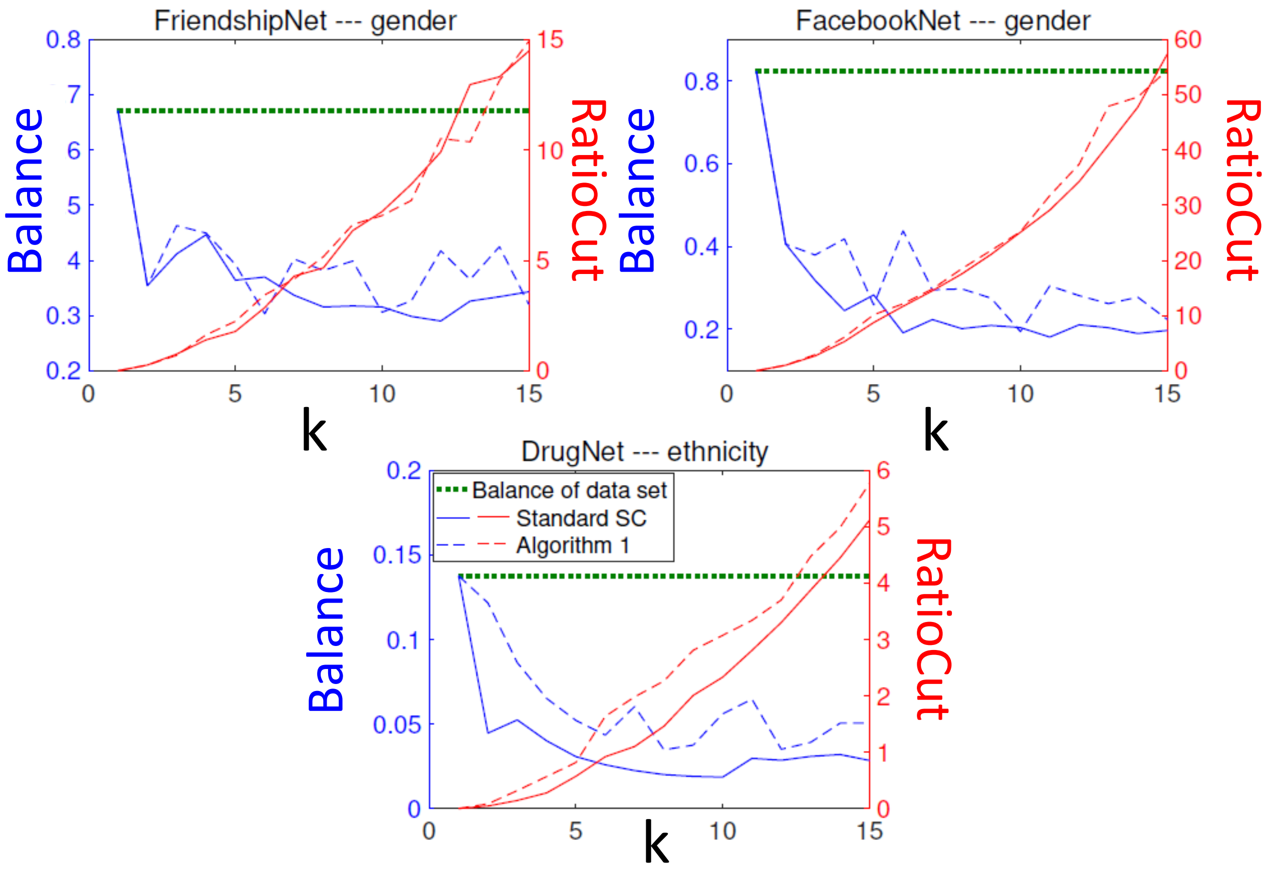
    for some $a > b > c > d$
  - A fair ground-truth clustering $V = C_1 \cup C_2$
- **Theorem:** Fair SC recovers the ground-truth clustering $C_1 \cup C_2$
- **Example**
  - Standard SC is likely to return $V_1 \cup V_2$
  - Fair SC will return $C_1 \cup C_2$



[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.

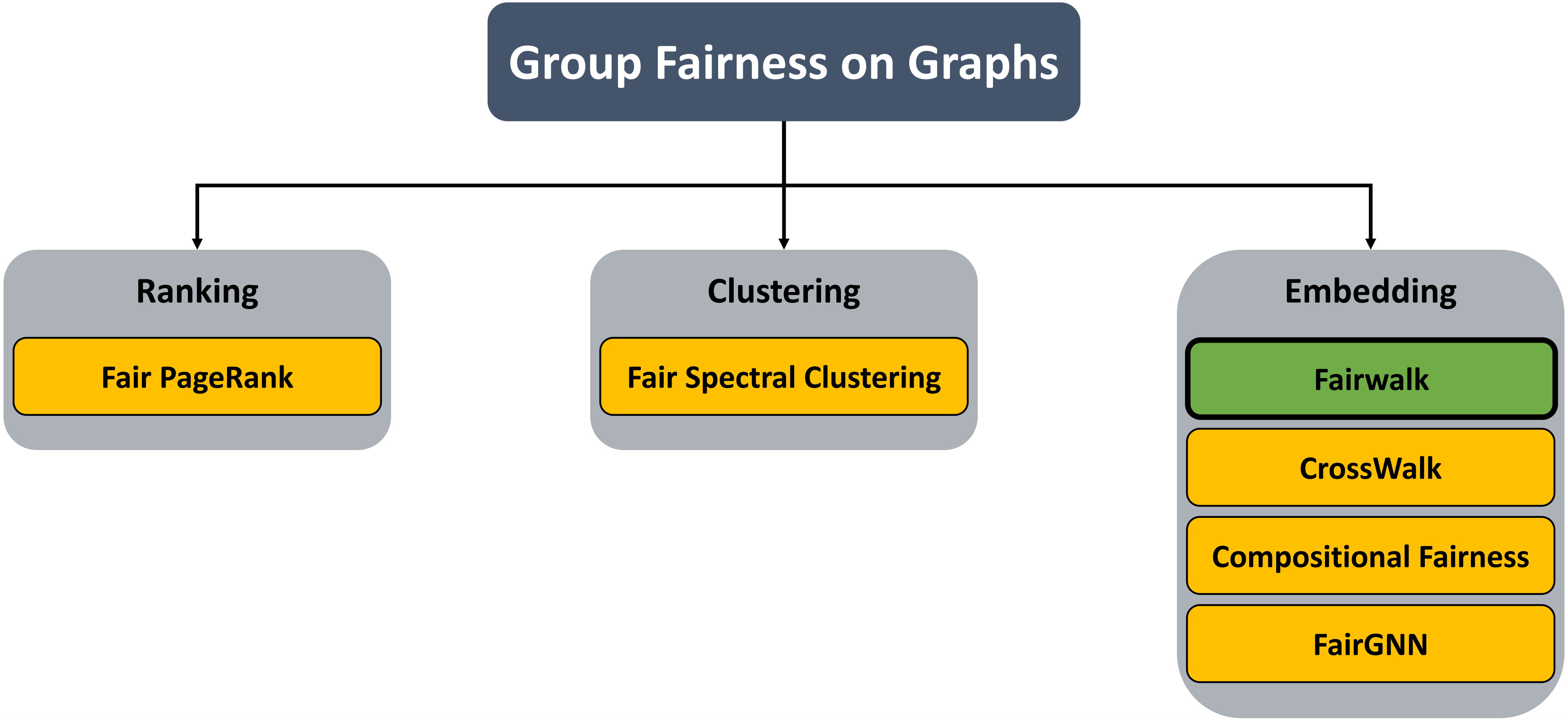
Fair Spectral Clustering: Experiment

- **Observation:** Fairer (higher balance score) with similar ratio cut values for the proposed method (Algorithm 1 in the figure)



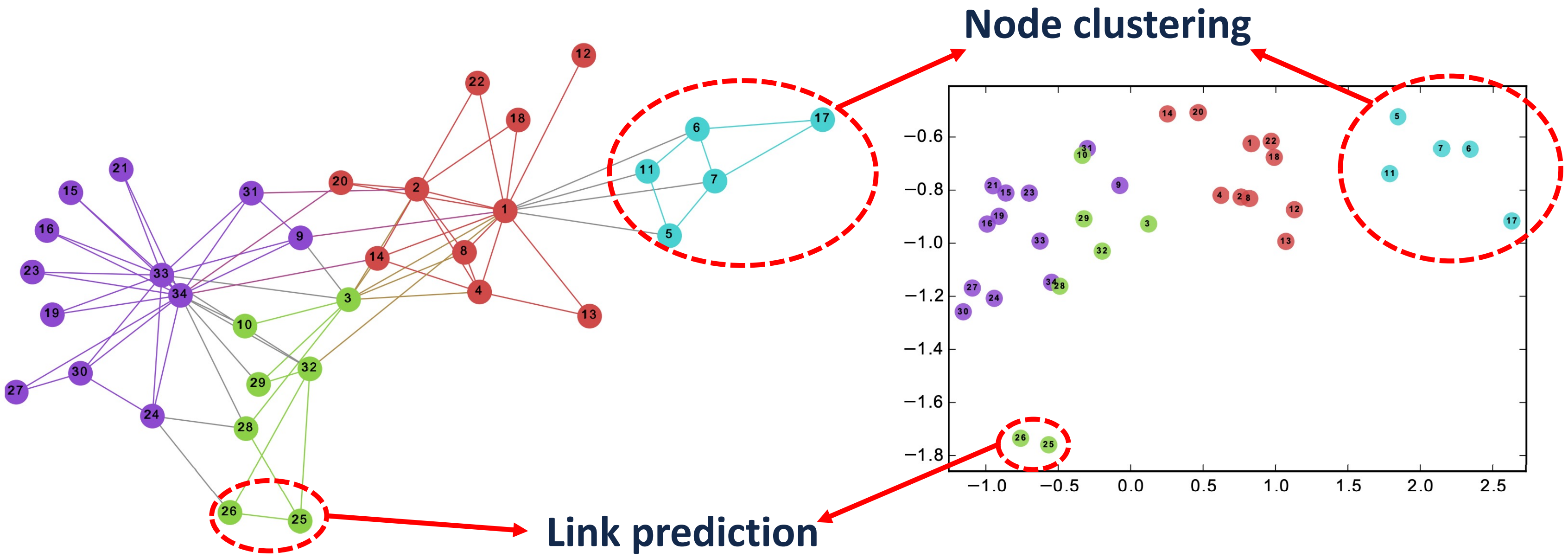
[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.

Overview of Part I



Preliminary: Node Embedding

- **Motivation:** learn low-dimensional node representations that preserve structural/attributive information
- **Applications**
  - – Node classification
  - – Link prediction
  - – Node visualization



(a) Input: Karate Graph (b) Output: Representation

Visualization of Node Embedding

[1] Perozzi, B., Al-Rfou, R., & Skiena, S.. Deepwalk: Online Learning of Social Representations. KDD 2014.
[2] Grover, A., & Leskovec, J.. node2vec: Scalable Feature Learning for Networks. KDD 2016.
[3] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O.. Translating Embeddings for Modeling Multi-relational Data. NeurIPS 2013.

Preliminary: Setup of Node Embedding

- **Two key components:** pairwise scoring function + loss function
- **Pairwise scoring function**
  - Suppose a node pair $e = (u, v)$; $\mathbf{z}_u$ is embedding of u;
  - **Dot product:** $s(e) = s(\langle \mathbf{z}_u, \mathbf{r}, \mathbf{z}_v \rangle) = \mathbf{z}_u^T \mathbf{z}_v$
  - **TransE:** $s(e) = s(\langle \mathbf{z}_u, \mathbf{r}, \mathbf{z}_v \rangle) = -\|\mathbf{z}_u + \mathbf{r} - \mathbf{z}_v\|_2^2$
- **Pairwise loss function**
  - Suppose e_i^- is i-th negative sample for node pair $e = (u, v)$
  - **Skip-gram loss**

$$L_e\left(s(e), s(e_1^-), \dots, s(e_m^-)\right) = -\log\left[\sigma\left(s(e)\right)\right] - \sum_{i=1}^{m} \log\left[1 - \sigma\left(s(e_i^-)\right)\right]$$

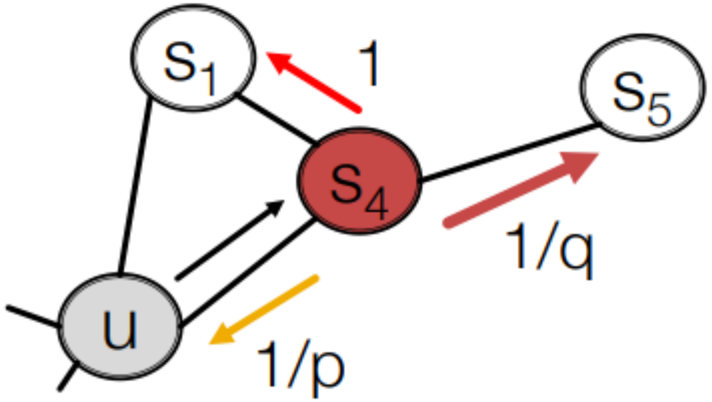
  - **Max-margin loss**

$$L_e\left(s(e), s(e_1^-), \dots, s(e_m^-)\right) = \sum_{i=1}^{m} \max(1 + s(e) - s(e_i^-), 0)$$

[1] Perozzi, B., Al-Rfou, R., & Skiena, S.. Deepwalk: Online Learning of Social Representations. KDD 2014.
[2] Grover, A., & Leskovec, J.. node2vec: Scalable Feature Learning for Networks. KDD 2016.
[3] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O.. Translating Embeddings for Modeling Multi-relational Data. NeurIPS 2013.

Preliminary: Random Walk-based Node Embedding

- **Goal:** learn node embeddings that are predictive of nodes in its neighborhood
- **Key idea**
  - Simulate random walk as a sequence of nodes
  - Apply skip-gram technique to predict the context node
- **Example**
  - **DeepWalk:** random walk for sequence generation
  - **Node2vec:** biased random walk for sequence generation
    - **Return parameter** p: how fast the walk **explores** the neighborhood of the starting node
    - **In-out parameter** q: how fast the walk **leaves** the neighborhood of the starting node



[1] Perozzi, B., Al-Rfou, R., & Skiena, S.. Deepwalk: Online Learning of Social Representations. KDD 2014.
[2] Grover, A., & Leskovec, J.. node2vec: Scalable Feature Learning for Networks. KDD 2016.

Fairness Measure: Statistical Parity

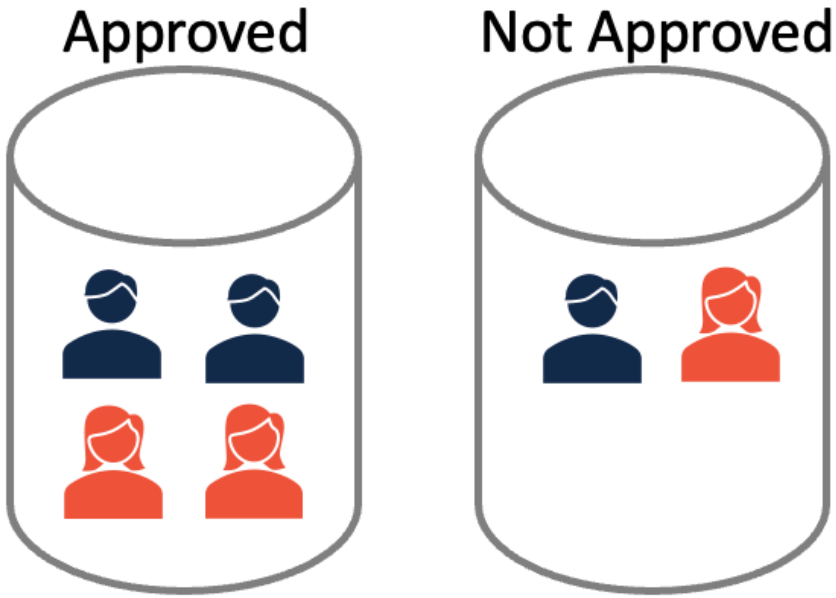
- **Statistical parity**
  - **Given:** (1) a sensitive attribute $\mathcal{S}$; (2) multiple demographic groups $\mathcal{G}^{\mathcal{S}}$ partitioned by $\mathcal{S}$

  Extension to multiple groups: variance among the acceptance rates of each group in $\mathcal{G}^{\mathcal{S}}$

$$\text{bias}^{\text{SI}}(\mathcal{G}^{\mathcal{S}}) = \text{Var}\left(\left\{\text{acceptance}-\text{rate}\left(G^{\mathcal{S}}\right) \middle| G^{\mathcal{S}} \in \mathcal{G}^{\mathcal{S}}\right\}\right)$$

- **Example:** a network of three and three
  - acceptance−rate( )=2/3
  - acceptance−rate( )=2/3
  - $\text{bias}^{\text{SI}} = \text{Var}\left(\left\{\frac{2}{3}, \frac{2}{3}\right\}\right) = 0$

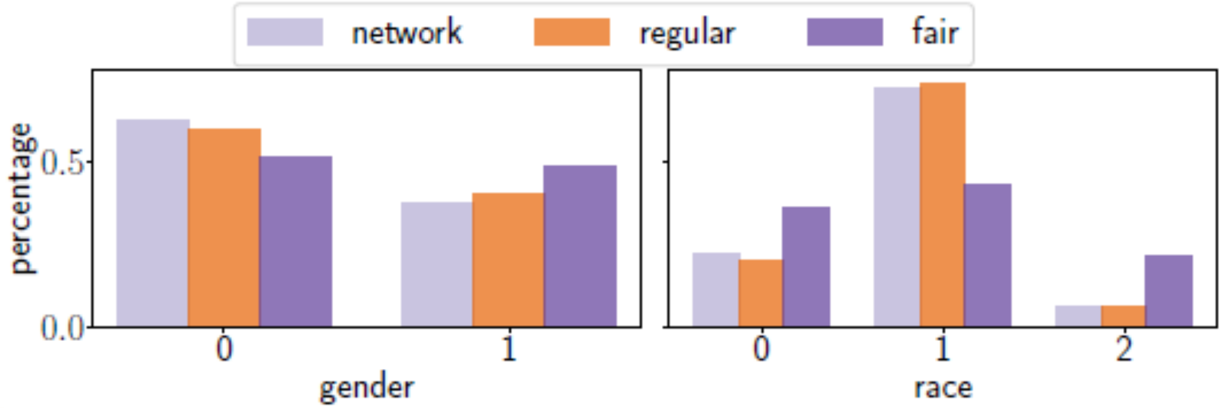
Fair result
Zero bias between male and female



[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.
[2] Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.

Fairwalk: Solution

- **Key idea:** modify the random walk procedure in node2vec
- **Steps of Fairwalk**
  - Partition neighbors into demographic groups
  - Uniformly sample a demographic group to walk to
  - Randomly select a neighboring node within the chosen demographic group
- **Example:** ratio of each demographic group
  - Original network vs. regular random walk vs. fair random walk



[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.

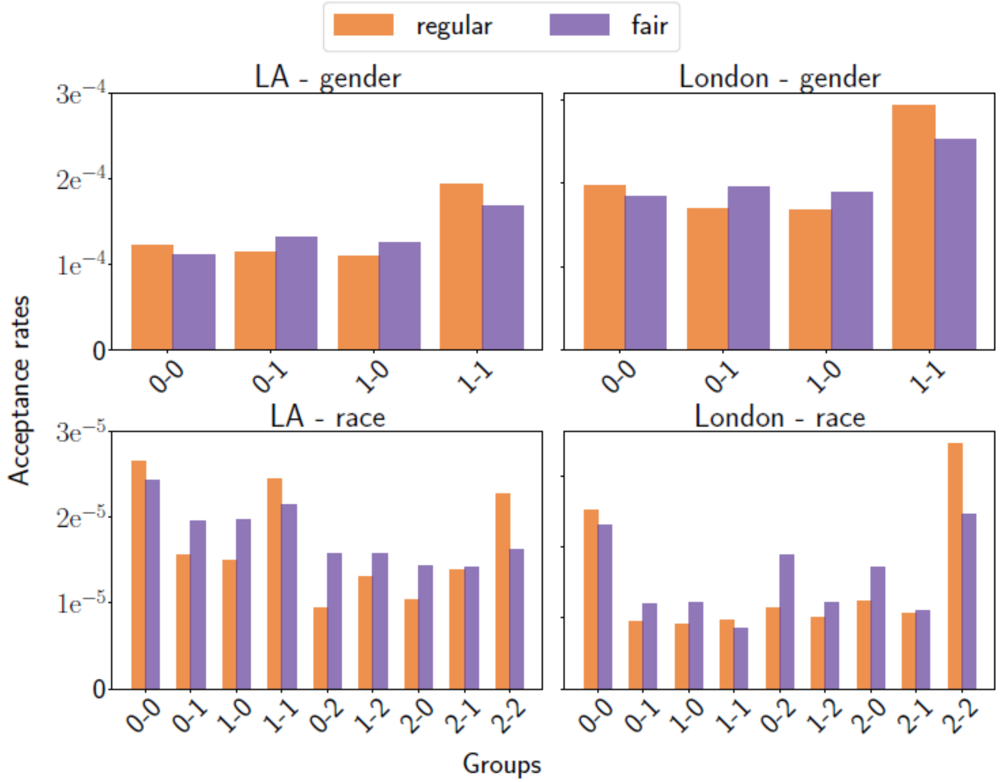
Fairwalk vs. Existing Works

- **Fairwalk vs. node2vec**
  - **Node2vec:** skip-gram model + walk sequences by original random walk
  - **Fairwalk:** skip-gram model + walk sequences by fair random walk
- **Fairwalk vs. fairness-aware PageRank**
  - **Fairness-aware PageRank:** the minority group should have a certain proportion of PageRank probability mass
  - **Fairwalk:** all demographic group have the same random walk transition probability mass

[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.
[2] Grover, A., & Leskovec, J.. node2vec: Scalable Feature Learning for Networks. KDD 2016.
[3] Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.

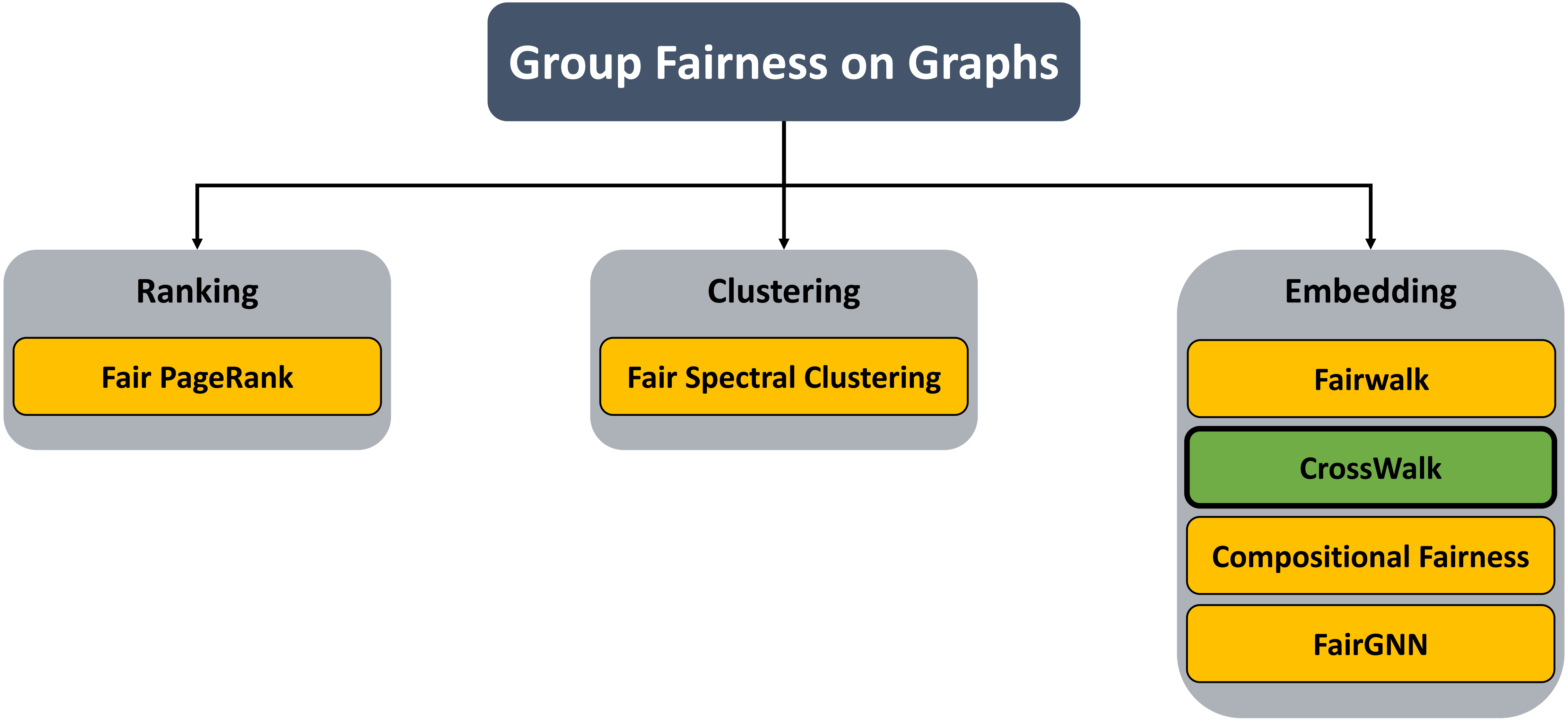
Fairwalk: Results on Statistical Parity

- **Observations**
  - Fairwalk achieves a more balanced acceptance rates among groups
  - Fairwalk increases the fraction of cross-group recommendations



[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.

Overview of Part I



Recap: Statistical Parity in Fairwalk

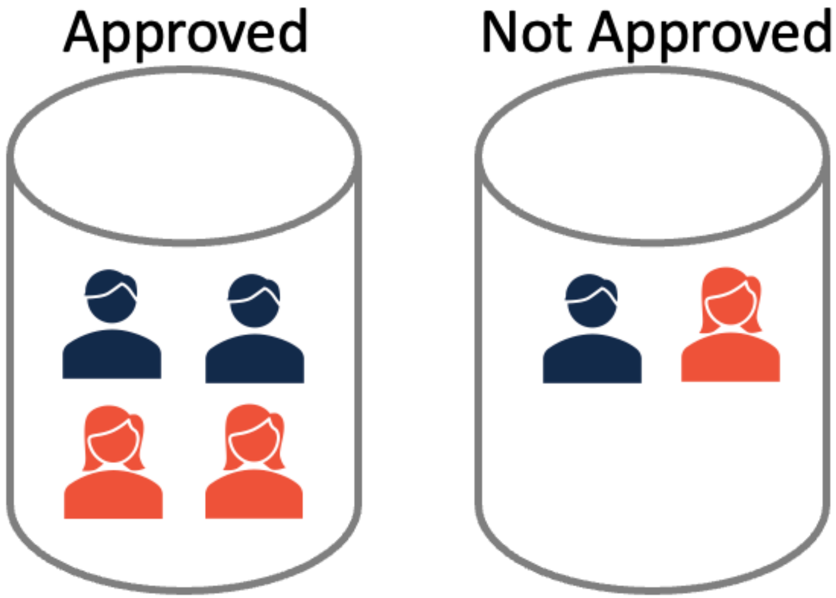
- **Statistical parity**
  - **Given:** (1) a sensitive attribute $\mathcal{S}$; (2) multiple demographic groups $\mathcal{G}^{\mathcal{S}}$ partitioned by $\mathcal{S}$

  Extension to multiple groups: variance among the acceptance rates of each group in $\mathcal{G}^{\mathcal{S}}$

$$\text{bias}^{\text{SI}}(\mathcal{G}^{\mathcal{S}}) = \text{disparity} = \text{Var}\left(\left\{\text{acceptance}-\text{rate}\left(G^{\mathcal{S}}\right) \middle| G^{\mathcal{S}} \in \mathcal{G}^{\mathcal{S}}\right\}\right)$$

- **Example:** a network of three and three
  - $\text{acceptance}-\text{rate}( ) = 2/3$
  - $\text{acceptance}-\text{rate}( ) = 2/3$
  - $\text{bias}^{\text{SI}} = \text{Var}\left(\left\{\frac{2}{3}, \frac{2}{3}\right\}\right) = 0$

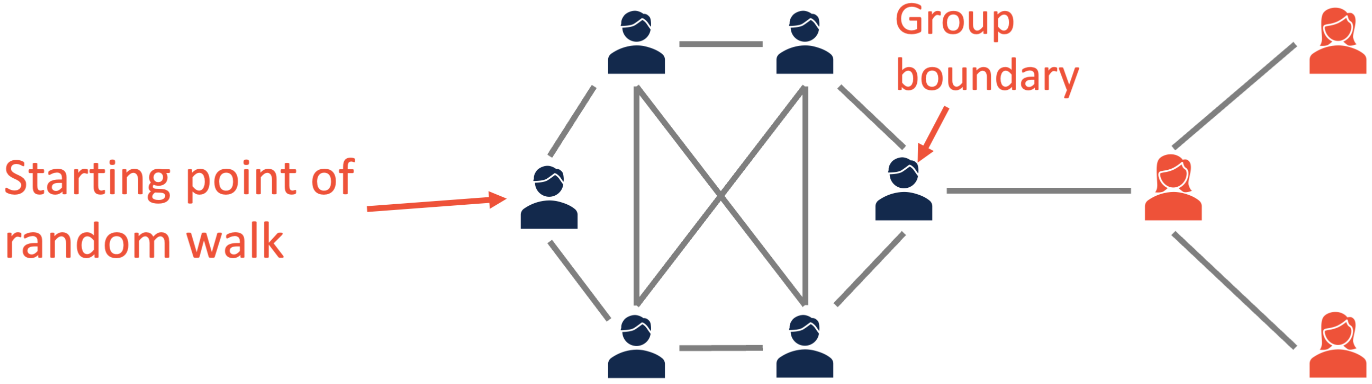
Fair result
Zero bias between male and female



[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.

Limitations: Fairwalk

- **Steps of Fairwalk**
  - Partition neighbors into demographic groups
  - Uniformly sample a demographic group to walk to
  - Randomly select a neighboring node within the chosen demographic group
- **Example:** what if all neighbors belong to the same group?

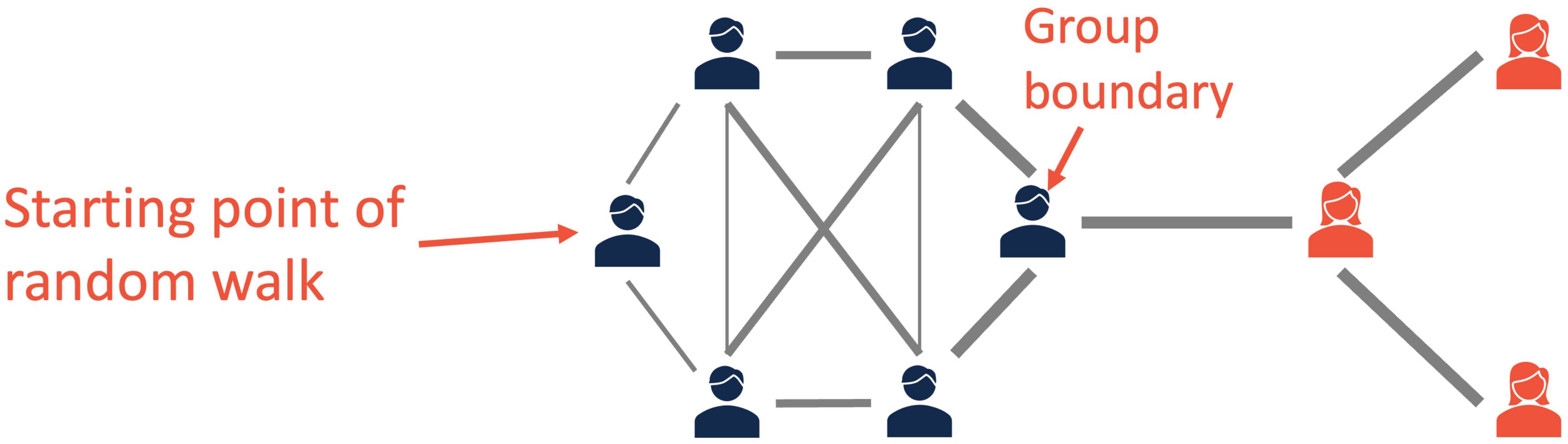


- **Observation:** Fairwalk may get trapped into the majority group
- **Question:** how to let the walker go to group boundary and go across group more often?

[1] Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.

CrossWalk: Key Idea

- **Key idea:** upweight edges whose target nodes are either
  – Closer to group boundary
  – Not in the same demographic group as source node
- **Example**
  – Edge strength is proportional to the transition probability



[1] Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.

CrossWalk: Proximity to Group Boundary

- **Intuition:** assign higher weight to edges whose target nodes are closer to group boundary
- **Solution:** the proximity $m(u)$ of node u can be calculated by
  - Performing a fixed-length random walk (length = d) r times
  - Calculating the probability that it walks to a node in another demographic group
- **Example:** suppose we have 2 random walks of length 5 for a node 

Walk #1:
Walk #2:

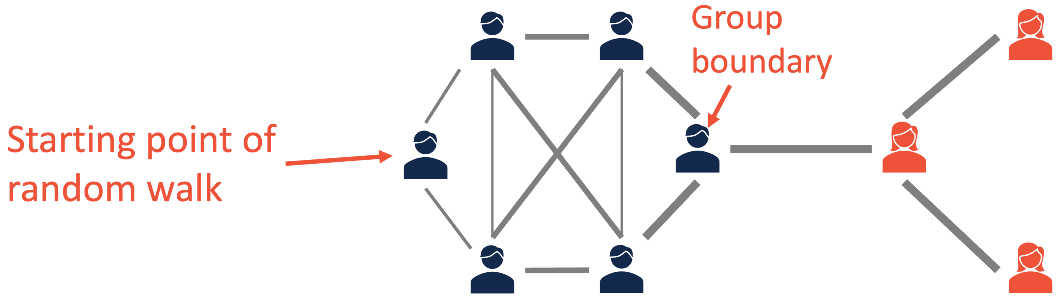
$m($$) = \frac{1+2}{2\times5} = 0.3$

- **Proximity-aware edge reweighting**
  - w_{uv}: original edge weight between node u and node v
  - $\mathcal{N}_u$: neighborhood of node u
  - p: a hyperparameter

$$w'_{uv} \propto \frac{m(v)^p}{\sum_{z\in\mathcal{N}_u} w_{uz} m(z)^p} w_{uv}$$

[1] Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.

CrossWalk: Solution

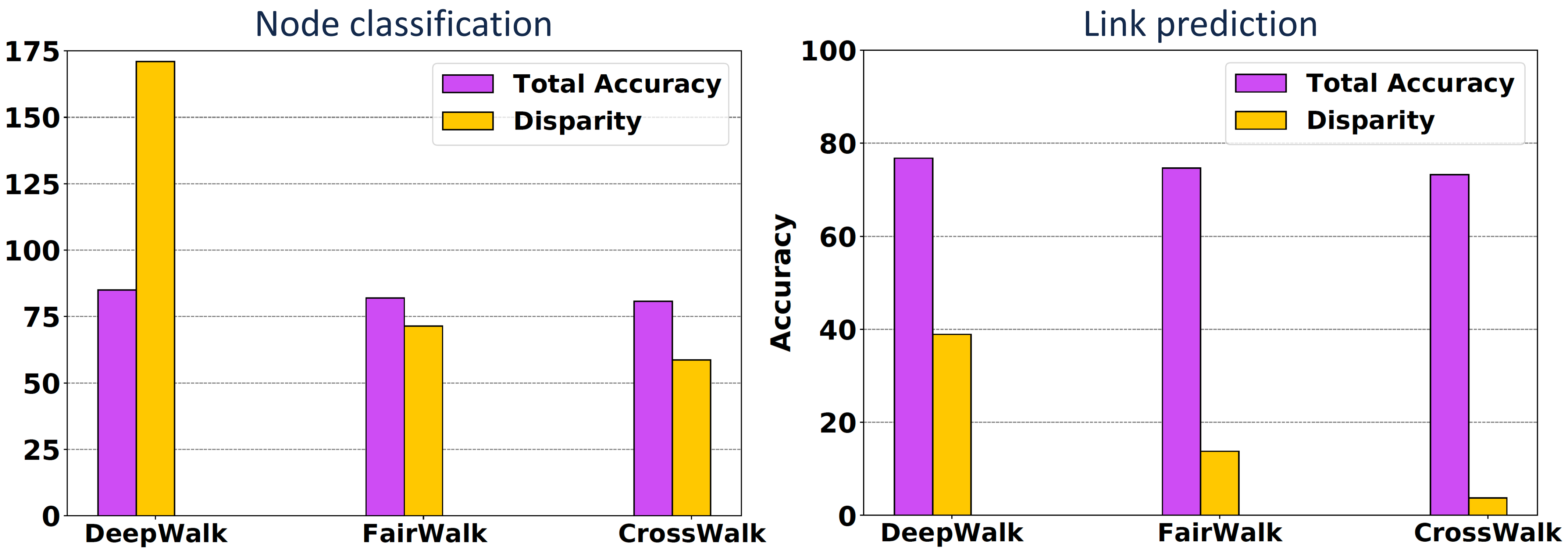


- **Given**
  - α: a hyperparameter to control within-group/cross-group probability
  - $\mathcal{N}_u$: node u's neighborhood
  - $|R_u|$: number of different demographic groups in $\mathcal{N}_u$
- **Edge reweighting:** for a node u and its neighbor $v, \forall v \in \mathcal{N}_u$
  - u and v are in the same group: $w'_{uv} = (1-\alpha)\frac{m(v)^p}{\sum_{z\in\mathcal{N}_u} w_{uz} m(z)^p} w_{uv}$
  - u and v are NOT in the same group: $w'_{uv} = \frac{\alpha}{|R_u|}\frac{m(v)^p}{\sum_{z\in\mathcal{N}_u} w_{uz} m(z)^p} w_{uv}$
- **Key steps**
  - Generate biased random walk sequences using the reweighted edges
  - Learn node representations using skip-gram based techniques on the biased random walk sequences

[1] Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.

CrossWalk: Experiment

- **Observation:** CrossWalk achieves a comparable performance in accuracy with a much smaller bias



[1] Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.

Overview of Part I

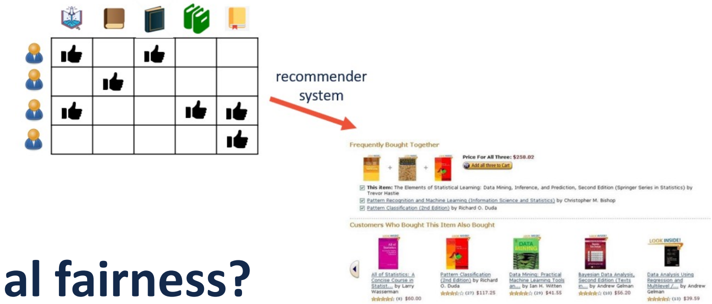
Group Fairness on Graphs

- **Ranking**
  - Fair PageRank
- **Clustering**
  - Fair Spectral Clustering
- **Embedding**
  - Fairwalk
  - CrossWalk
  - Compositional Fairness
  - FairGNN

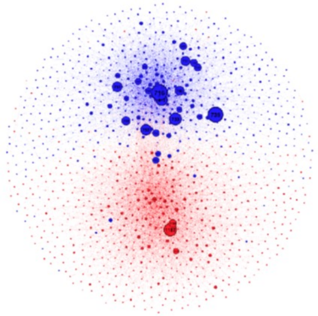
Compositional Fairness in Node Embedding

- **Why fairness for embeddings?**
  - Not just one classification task that considers fairness (e.g., ranking, clustering)

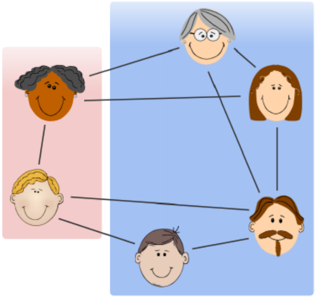
recommendation



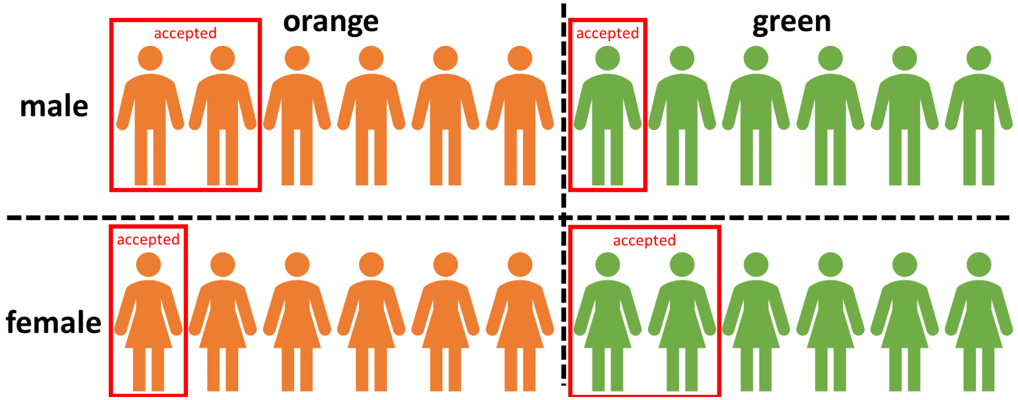
ranking



clustering



- **Why compositional fairness?**
  - **Compositional fairness:** accommodation to a combination of sensitive attributes
  - Often many possible sensitive attributes for a downstream task



- **Gender:** male vs. female
- **Race*:** orange vs. green

* We use imaginary race groups to avoid potential offenses

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Fairness Measure: Representational Invariance

- **Intuition:** independence between the learned embedding $\mathbf{z}$ and a sensitive attribute a

$$\mathbf{z}_u \perp a_u, \forall \text{ node } u$$

where a_u is the sensitive value of node u

- **Formulation:** mutual information minimization

$$I(\mathbf{z}_u, a_u) = 0, \forall \text{ node } u$$

  - Analogous to statistical parity in classification task
  - **Key idea:** fail to predict a_u using $\mathbf{z}_u$

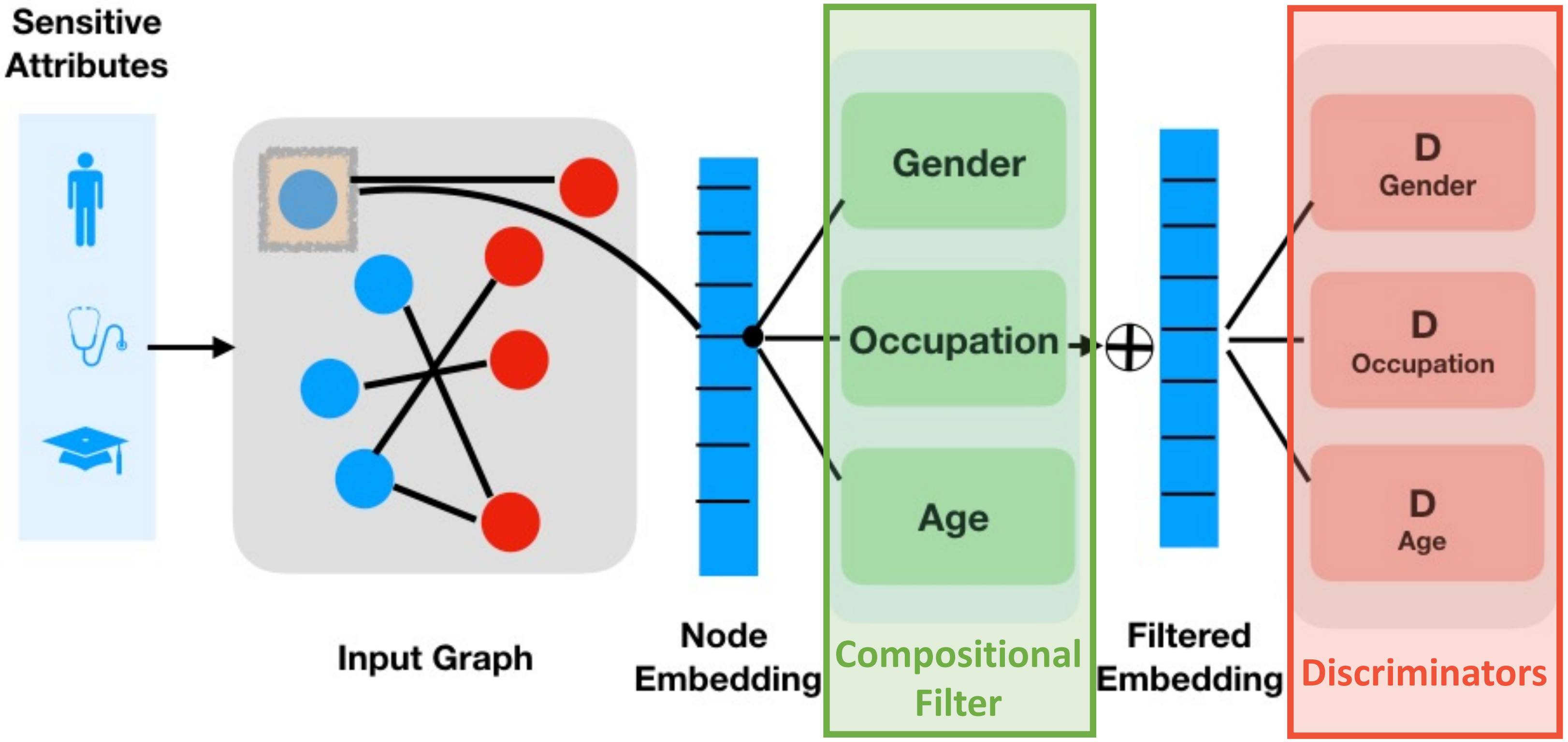
Corresponding to 'adversarial'

- **Solution:** adversarial learning
  - Maximize the error to predict sensitive feature

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Compositional Fairness: Framework

- **Overview:** the proposed compositional fairness framework
- **Key components:** (1) Compositional Filter (C-ENC) and (2) Discriminators (D_k)



[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Key Component #1: Compositional Filter

(Also called compositional encoder, i.e., C-ENC)

- **Goal:** filter sensitive information from the embeddings
  - The 'filtered' embedding should be invariant to those attributes
- **Formulation**

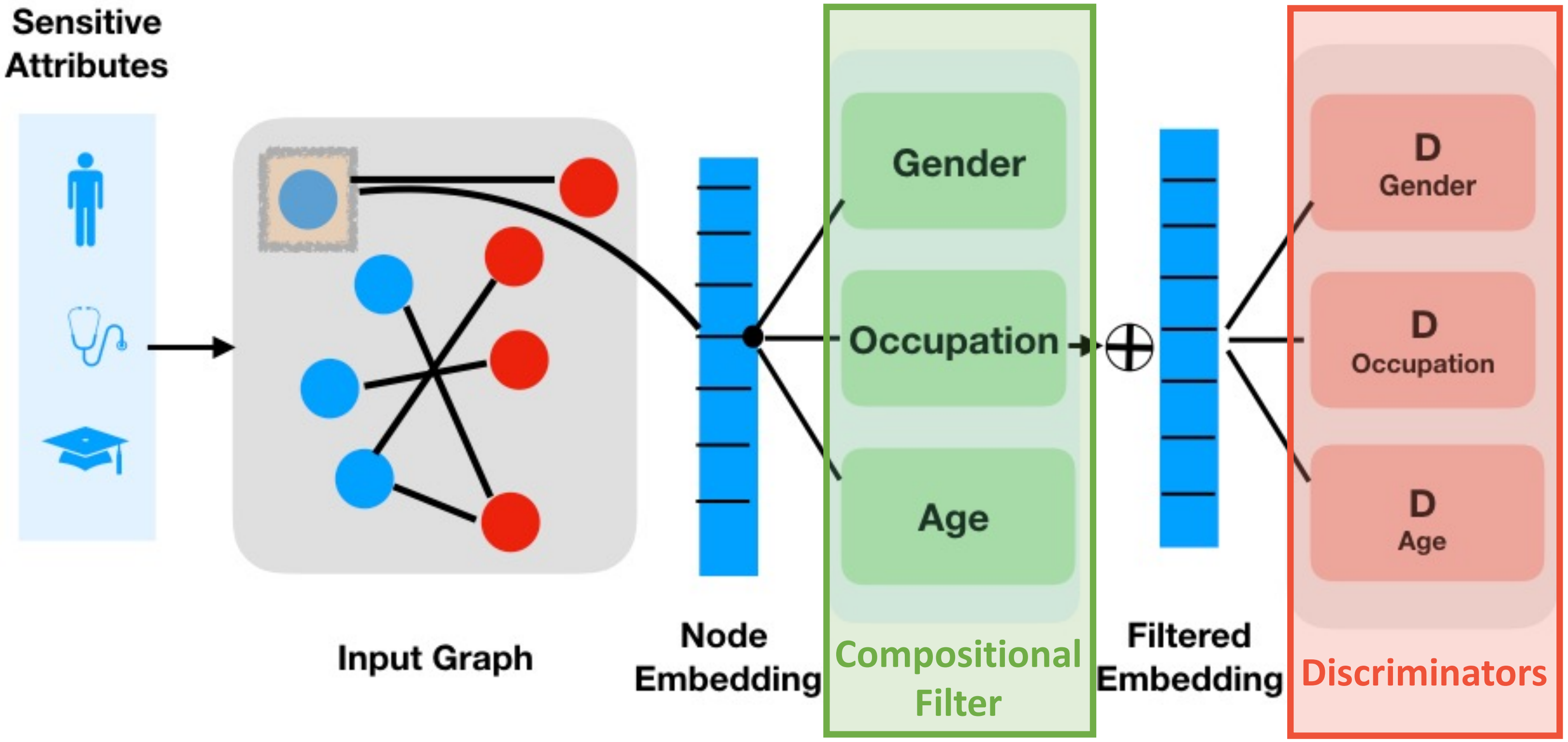
$$\text{C}-\text{ENC}(u, S) = \frac{1}{|S|} \sum_{k \in S} f_k\big(\text{ENC}(u)\big)$$

  - Compositional filter: a collection of filters
  - Filter: trainable function f_k (neural networks, e.g., MLP)
  - Input: node ID u and the set of sensitive attributes S (e.g., gender, age)
  - Compositionality: summation over all sensitive attributes

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Compositional Fairness: Framework

- **Overview:** the proposed compositional fairness framework
- **Key components:** (1) Compositional Filter (C-ENC) and (2) Discriminators (D_k)



[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Key Component #2: Discriminator

- **Goal:** predict the sensitive attribute from the 'filtered' embeddings
- **Formulation**

$$\mathrm{D}_k\left(\mathrm{C{-}ENC}(u,S), a^k\right) = \Pr\left(a_u = a^k \middle| \mathrm{C-ENC}(u,S)\right)$$

  - D_k: discriminator for k-th sensitive attribute
  - Input: node u's 'filtered' embedding and attribute value
  - $\Pr\left(a_u = a^k \middle| \mathrm{C-ENC}(u,S)\right)$: likelihood that node u has that attribute value

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

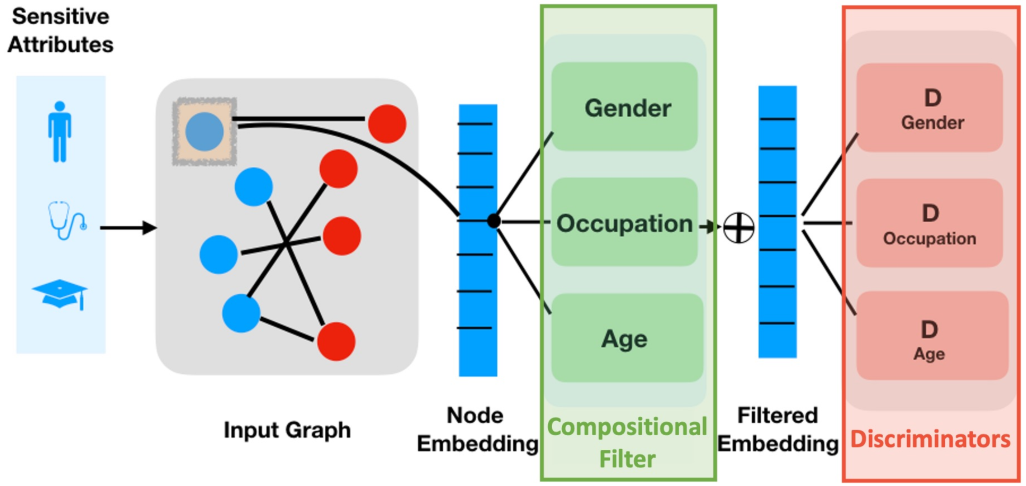
Compositional Fairness: Loss Function

- **Pairwise loss function**

$$L(e) = L_{\text{edge}}\left(s(e), s(e_1^-), \dots, s(e_m^-)\right) + \lambda \sum_{k\in S} \sum_{a^k \in \mathcal{A}_k} \log\left(\mathrm{D}_k\left(\mathrm{C\text{-}ENC}(u,S), a^k\right)\right)$$

  - L_{edge}: pairwise loss function for graph embedding
  - $\log\left(\mathrm{D}_k\left(\mathrm{C\text{-}ENC}(u,S), a^k\right)\right)$: the discriminator fails to predict sensitive attribute correctly with the 'filtered' embeddings

- **Advantages**
  - Simple intuition
  - Flexible and easy-to-implement module
  - Plug-and-play style



[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Compositional Fairness: Fairness Results

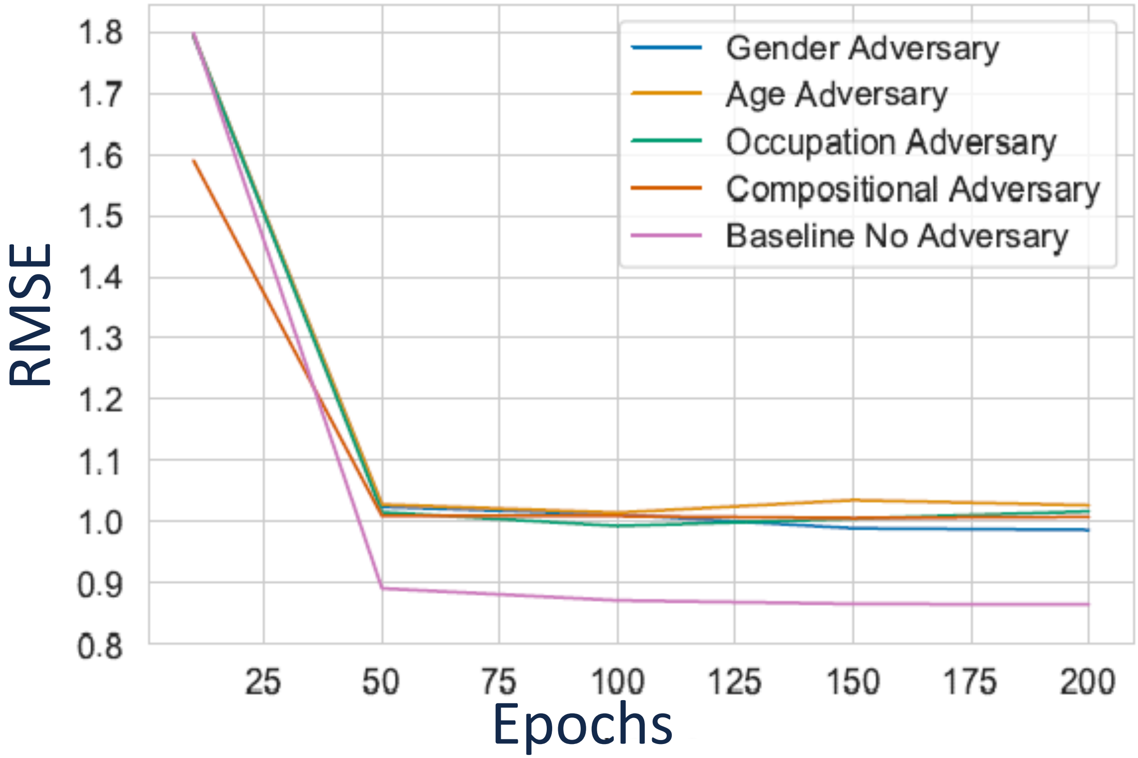
- **Task:** classifying the sensitive attribute from the learned node embeddings
  - **Baseline methods:** each adversary is a 2-layer MLP
    - Baseline (no adversary): Vanilla model train without fairness consideration
    - Independent adversary: independent adversarial model for each attribute
    - Compositional adversary: The proposed full compositional model
- **Observations**
  - Accuracy of compositional adversary is no better than majority classifier
  - Performance of compositional adversary is at the same level with independent adversaries

MovieLens1M	Baseline No Adversary	Gender Adversary	Age Adversary	Occupation Adversary	Comp. Adversary	Majority Classifier	Random Classifier	
Gender	0.712	0.532	0.541	0.551	0.511	0.5	0.5	AUC
Age	0.412	0.341	0.333	0.321	0.313	0.367	0.141	Micro F1
Occupation	0.146	0.141	0.108	0.131	0.121	0.126	0.05	

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Compositional Fairness: Effectiveness Results

- **Task:** recommendation
- **Observation:** there is only a small increase in root mean squared error (RMSE) compared with the vanilla model



[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Overview of Part I

Group Fairness on Graphs

- **Ranking**
  - Fair PageRank
- **Clustering**
  - Fair Spectral Clustering
- **Embedding**
  - Fairwalk
  - CrossWalk
  - Compositional Fairness
  - FairGNN

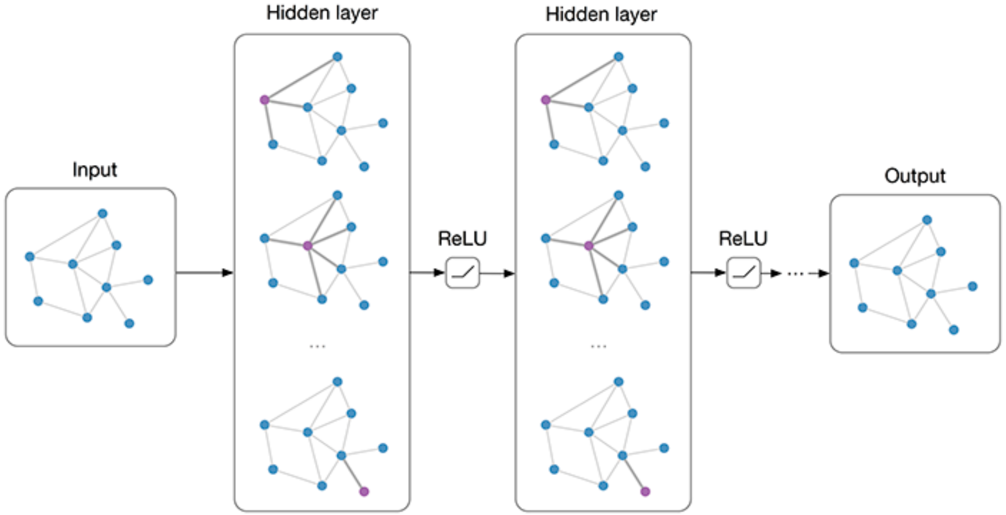
Preliminary: Graph Neural Network (GNN)

- **Key idea:** learn node representations by aggregating information from the neighbors
- **Formulation**

Node representation of node i · Weight parameters · Neighborhood aggregation function · Local neighborhood of node i

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \mathrm{AGG}\left(\mathbf{h}_j^{(l)}, \forall j \in \mathcal{N}(i)\right)\right)$$

  - **GCN:** $\mathrm{AGG}\left(\mathbf{h}_j^{(l)}, \forall j \in \mathcal{N}(i)\right) = \sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \mathbf{h}_j^{(l)}$
    - $a_{ij} = \frac{1}{\sqrt{d_i+1}\sqrt{d_j+1}}$: weight of the edge between node i w.r.t. node j
    - d_i, d_j: degree of node i and node j, respectively
  - **GAT:** $\mathrm{AGG}\left(\mathbf{h}_j^{(l)}, \forall j \in \mathcal{N}(i)\right) = \sum_{j \in \mathcal{N}_i \cup \{i\}} b_{ij} \mathbf{h}_j^{(l)}$
    - b_{ij}: self attention weight of node i w.r.t. node j
  - **GraphSAGE:** $\mathrm{AGG}\left(\mathbf{h}_j^{(l)}, \forall j \in \mathcal{N}(i)\right) = \mathbf{h}_i^{(l)} || \left(\sum_{j \in \mathcal{N}_i} c_{ij} \mathbf{h}_j^{(l)}\right)$
    - $c_{ij} = \frac{1}{d_i+1}$: weight of the edge between node i w.r.t. node j
    - $||$: concatenation operation

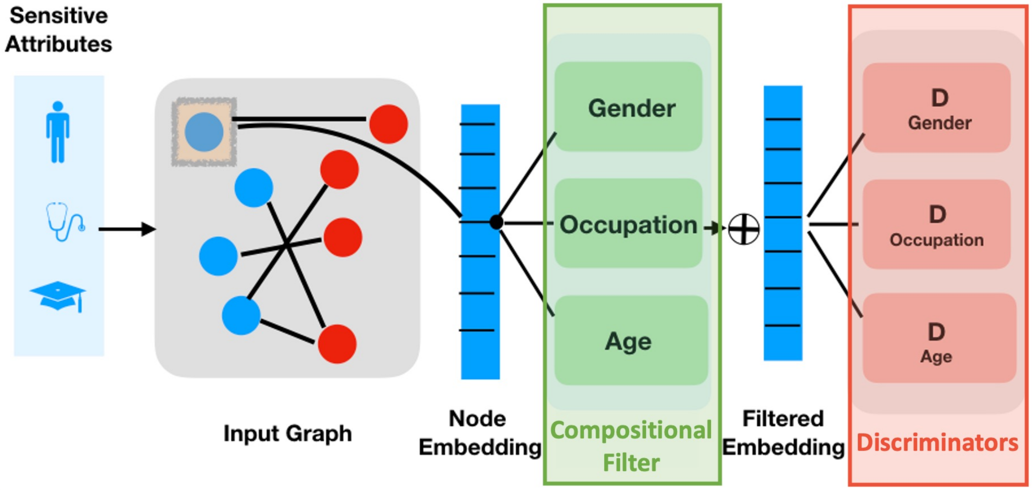


- **Applications:** node classification, link prediction, …

[1] Kipf, T. N., & Welling, M.. Semi-supervised Classification with Graph Convolutional Networks. ICLR 2017.
[2] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y.. Graph Attention Networks. ICLR 2018.
[3] Hamilton, W., Ying, Z., & Leskovec, J.. Inductive Representation Learning on Large Graphs. NeurIPS 2017.

Preliminary: Adversarial Debiasing

- **Key idea:** learn node representations that
  - Preserve structural/attributive information
  - Fail to predict sensitive attribute of the corresponding nodes
- **Solution:** adversarial learning-based approach
  - Minimize a task-specific loss function to learn 'good' representations
  - Maximize the error of predicting sensitive feature to learn 'fair' representations
- **Example:** compositional fairness constraints (CFC) framework



[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

Limitation: Adversarial Learning-based Debiasing

- **Example:** compositional fairness framework

$$L(e) = L_{\text{edge}}\big(s(e), s(e_1^-), \dots, s(e_m^-)\big) + \lambda \sum_{k \in S} \sum_{a^k \in \mathcal{A}_k} \log\Big(\mathrm{D}_k\big(\mathrm{C{-}ENC}(u,S), a^k\big)\Big)$$

  - L_{edge}: pairwise loss function to learn 'good' embedding
  - $\log\Big(\mathrm{D}_k\big(\mathrm{C{-}ENC}(u,S), a^k\big)\Big)$: an adversary (a discriminator) to maximize the error of predicting sensitive attribute to learn 'fair' embedding

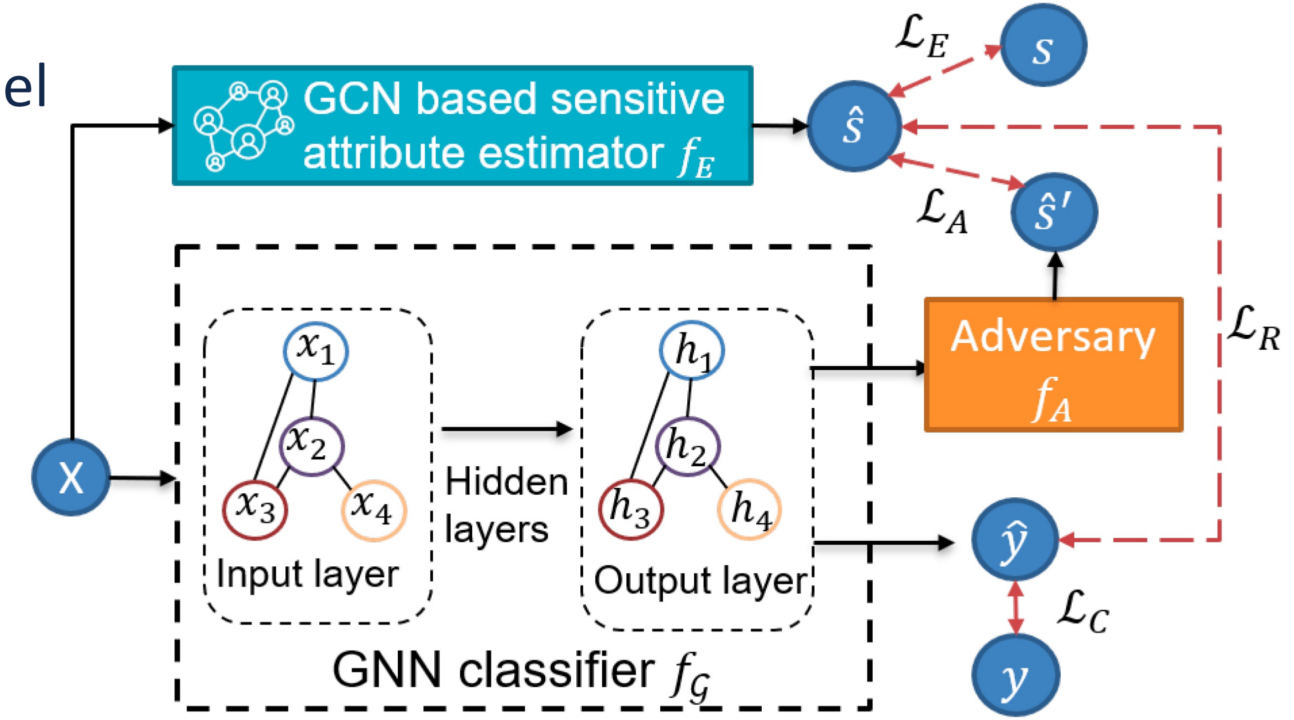
- **Limitations**
  - Require the sensitive attribute of many nodes to train a good discriminator
  - Ignore the fact that sensitive information is hard to obtain due to privacy

- **Question:** can we apply adversarial learning-based debiasing with limited sensitive attribute information?

[1] Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.

FairGNN: Fairness with Limited Sensitive Attribute Information

- **Key idea**
  - Train a sensitive attribute estimator to infer pseudo sensitive attribute
  - Train adversary to learn 'fair' embedding using the pseudo sensitive attribute
- **FairGNN framework**
  - GNN-based classifier to predict node label
    - Any GNN can be the backbone
  - Adversarial learning module to debias
    - GCN-based sensitive attribute estimator
    - Adversary
  - Covariance minimizer



[1] Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

FairGNN: Adversarial Debiasing Module

- **GCN-based sensitive attribute estimator**
  - **Intuition**: generate pseudo sensitive attribute for additional supervision
  - **Loss function**

$$\mathcal{L}_E = -\mathbb{E}_{u \in \mathcal{V}_S}[s_u \log \hat{s}_u]$$

    - s_u: ground-truth sensitive attribute information of node u
    - $\hat{s}_u$: predicted sensitive attribute information of node u
    - $\mathcal{V}_S$: a set of nodes with ground-truth sensitive attribute information
- **Adversary**
  - **Intuition:** maximize the error of predicting pseudo sensitive attribute information
  - **Loss function**

$$\mathcal{L}_A = \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\tilde{s}=1)}[\log f_A(\mathbf{h})] + \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\tilde{s}=0)}\left[\log\left(1 - f_A(\mathbf{h})\right)\right]$$

    - $\tilde{s}$: pseudo sensitive attribute information
    - $\mathbf{h} \sim p(\mathbf{h}|\tilde{s} = 1)$: randomly sample a node embedding whose corresponding node has $\tilde{s} = 1$
    - $f_A(\mathbf{h})$: output of

[1] Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

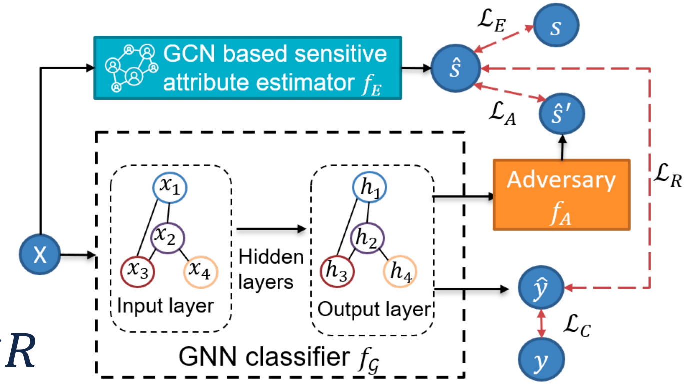
FairGNN: Covariance Minimizer

- **Observation:** adversarial learning is notoriously unstable to train
  - Failure to converge may cause discrimination
- **Key idea:** additional prerequisite of independence is needed to provide additional supervision signal
- **Solution:** absolute covariance between model prediction $\hat{y}$ and pseudo sensitive attribute $\hat{s}$ should be minimized
  - **Why absolute:** covariance can be negative

$$\mathcal{L}_R = |\mathrm{cov}(\hat{s}, \hat{y})| = |\mathbb{E}[(\hat{s} - \mathbb{E}[\hat{s}])(\hat{y} - \mathbb{E}[\hat{y}])]|$$

[1] Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

FairGNN: Overall Loss Function



- **Regularized learning**

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_E - \alpha\mathcal{L}_A + \beta\mathcal{L}_R$$

- **Intuition**
  - Minimize the classification loss $\mathcal{L}_C$ to learn representative node representation
  - Minimize the sensitive attribute estimation loss $\mathcal{L}_E$ to generate accurate pseudo sensitive attribute information
  - Maximize the adversarial loss $\mathcal{L}_A$ (i.e., $-\alpha\mathcal{L}_A$) to debias the learned node representation
  - Minimize the covariance $\mathcal{L}_R$ to stabilize the training of adversary

[1] Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

FairGNN: Experiment

- **Observation:** FairGNN achieves comparable node classification accuracy with a much smaller bias

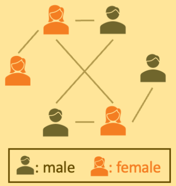
Dataset	Metrics	GCN	GAT	ALFR	ALFR-e	Debias	Debias-e	FCGE	FairGCN	FairGAT
Pokec-z	ACC (%)	70.2 ±0.1	70.4 ±0.1	65.4 ±0.3	68.0 ±0.6	65.2 ±0.7	67.5 ±0.7	65.9 ±0.2	**70.0 ±0.3**	**70.1 ±0.1**
	AUC (%)	77.2 ±0.1	76.7 ±0.1	71.3 ±0.3	74.0 ±0.7	71.4 ±0.6	74.2 ±0.7	71.0 ±0.2	**76.7 ±0.2**	**76.5 ±0.2**
	Δ_{SP} (%)	9.9 ±1.1	9.1 ±0.9	2.8 ±0.5	5.8 ±0.4	1.9 ±0.6	4.7 ±1.0	3.1 ±0.5	**0.9 ±0.5**	**0.5 ±0.3**
	Δ_{EO} (%)	9.1 ±0.6	8.4 ±0.6	1.1 ±0.4	2.8 ±0.8	1.9 ±0.4	3.0 ±1.4	1.7 ±0.6	**1.7 ±0.2**	**0.8 ±0.3**
Pokec-n	ACC (%)	70.5 ±0.2	70.3 ±0.1	63.1 ±0.6	66.2 ±0.5	62.6 ±0.9	65.6 ±0.8	64.8 ±0.5	**70.1 ±0.2**	**70.0 ±0.2**
	AUC (%)	75.1 ±0.2	75.1 ±0.2	67.7 ±0.5	71.9 ±0.3	67.9 ±0.7	71.7 ±0.7	69.5 ±0.4	**74.9 ±0.4**	**74.9 ±0.4**
	Δ_{SP} (%)	9.6 ±0.9	9.4 ±0.7	3.05 ±0.5	4.1 ±0.5	2.4 ±0.7	3.6 ±0.2	4.1 ±0.8	**0.8 ±0.2**	**0.6 ±0.3**
	Δ_{EO} (%)	12.8 ±1.3	12.0 ±1.5	3.9 ±0.6	4.6 ±1.6	2.6 ±1.0	4.4 ±1.2	5.5 ±0.9	**1.1 ±0.5**	**0.8 ±0.2**
NBA	ACC (%)	71.2 ±0.5	71.9 ±1.1	64.3 ±1.3	66.0 ±0.4	63.1 ±1.1	65.6 ±2.4	66.0 ±1.5	**71.1 ±1.0**	**71.5 ±0.8**
	AUC (%)	78.3 ±0.3	78.2 ±0.6	71.5 ±0.3	72.9 ±1.0	71.3 ±0.7	72.9 ±1.2	73.6 ±1.5	**77.0 ±0.3**	**77.5 ±0.7**
	Δ_{SP} (%)	7.9 ±1.3	10.2 ±2.5	2.3 ±0.9	4.7 ±1.8	2.5 ±1.5	5.3 ±0.9	2.9 ±1.0	**1.0 ±0.5**	**0.7 ±0.5**
	Δ_{EO} (%)	17.8 ±2.6	15.9 ±4.0	3.2 ±1.5	4.7 ±1.7	3.1 ±1.9	3.1 ±1.3	3.0 ±1.2	**1.2 ±0.4**	**0.7 ±0.3**

[1] Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

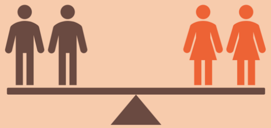
Coffee Break

- 15 minutes coffee break

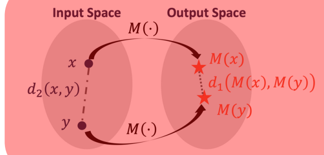
Roadmap



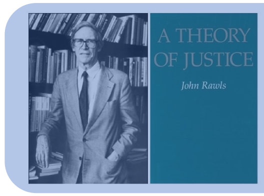
Introduction



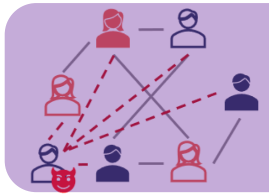
Part I: Group Fairness on Graphs



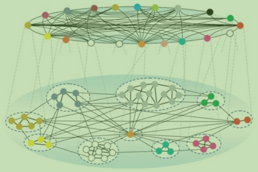
Part II: Individual Fairness on Graphs



Part III: Other Fairness on Graphs



Part IV: Beyond Fairness on Graphs



Part V: Future Trends

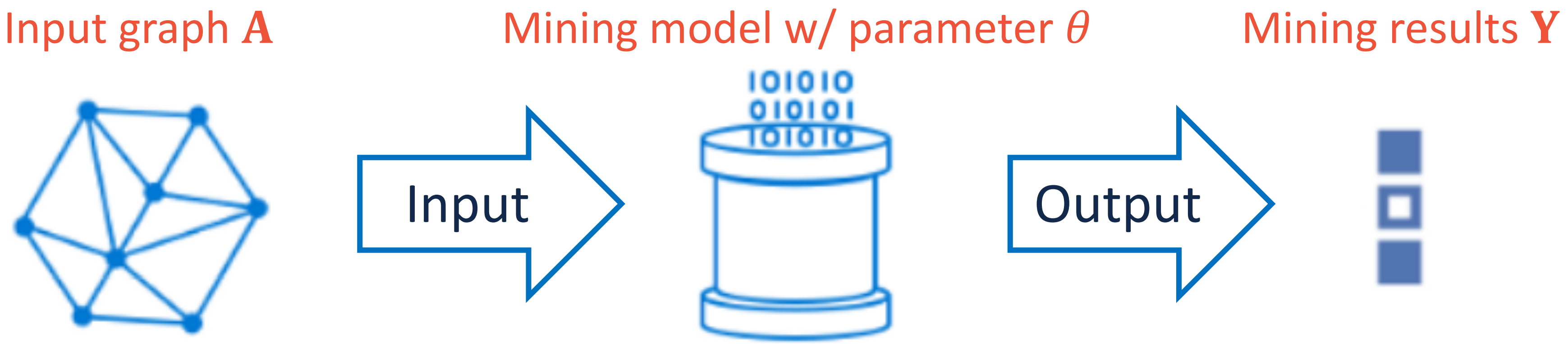
Overview of Part II

Individual Fairness on Graphs

- **Laplacian Regularizer**
  - **InFoRM**
- **Learning-to-Rank**
  - **REDRESS**

Graph Mining: An Optimization Perspective

- **A pipeline of graph mining**



- **Formulation**
  - **Input**
    - Input graph $\mathbf{A}$
    - Model parameters θ

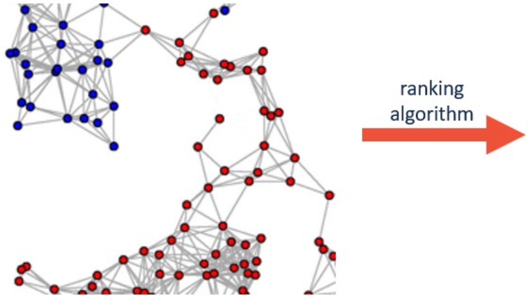
    Minimize task-specific loss function $l(\mathbf{A}, \mathbf{Y}, \theta)$
  - **Output:** mining results $\mathbf{Y}$
    - **Examples:** ranking vectors, class probabilities, embedding

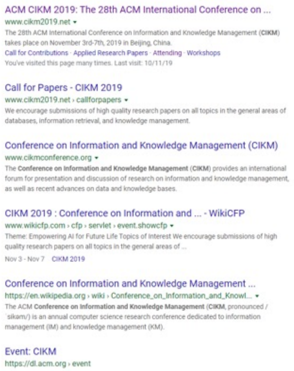
[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

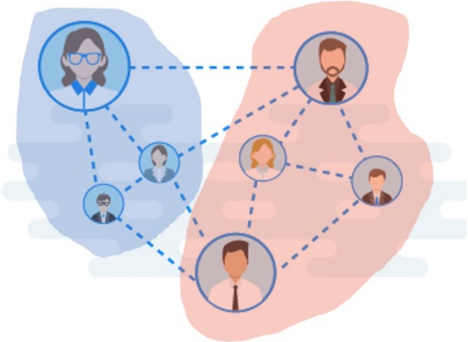
Classic Graph Mining Algorithms

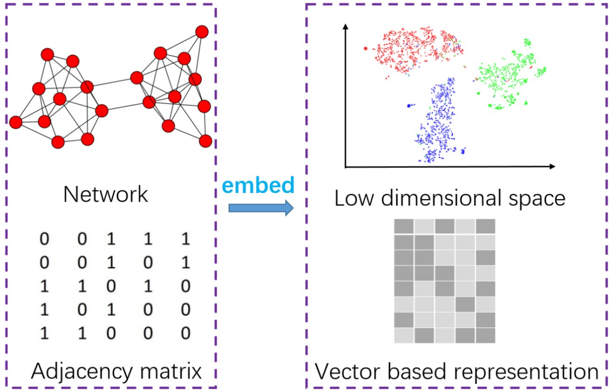
Examples of Classic Graph Mining Algorithm

Mining Task	Task Specific Loss Function $l()$	Mining Result Y^*	Parameters
PageRank	$\min_{\mathbf{r}} c\mathbf{r}^T(\mathbf{I}-\mathbf{A})\mathbf{r} + (1-c)\|\mathbf{r}-\mathbf{e}\|_F^2$	PageRank vector $\mathbf{r}$	damping factor c teleportation vector $\mathbf{e}$
Spectral Clustering	$\min_{\mathbf{U}} \mathrm{Tr}\,(\mathbf{U}^T\mathbf{L}\mathbf{U})$ $\text{s.t.}\ \ \mathbf{U}^T\mathbf{U}=\mathbf{I}$	eigenvectors $\mathbf{U}$	# clusters k
LINE (1st)	$\min_{\mathbf{X}} \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}[i,j]\left(\log g(-\mathbf{X}[j,:]\mathbf{X}[i,:]^T)\right)$ $+b\mathbb{E}_{j'\sim P_n}[\log g(-\mathbf{X}[j',:]\mathbf{X}[i,:]^T)]$	embedding matrix $\mathbf{X}$	embedding dimension d # negative samples b



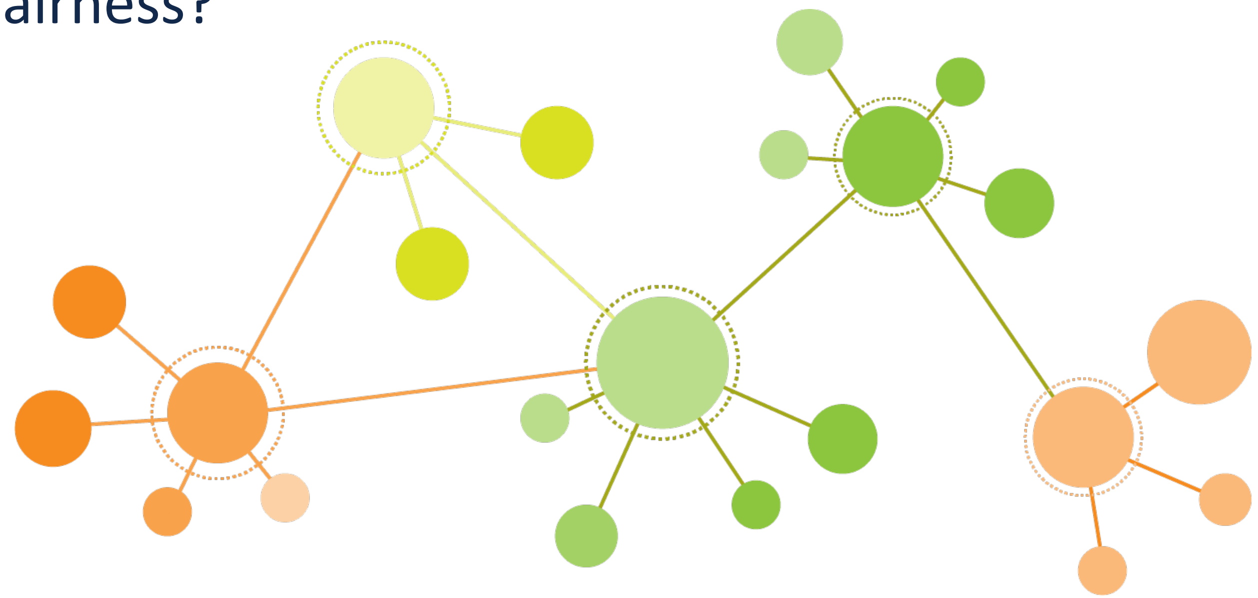






InFoRM: Individual Fairness on GRaph Mining

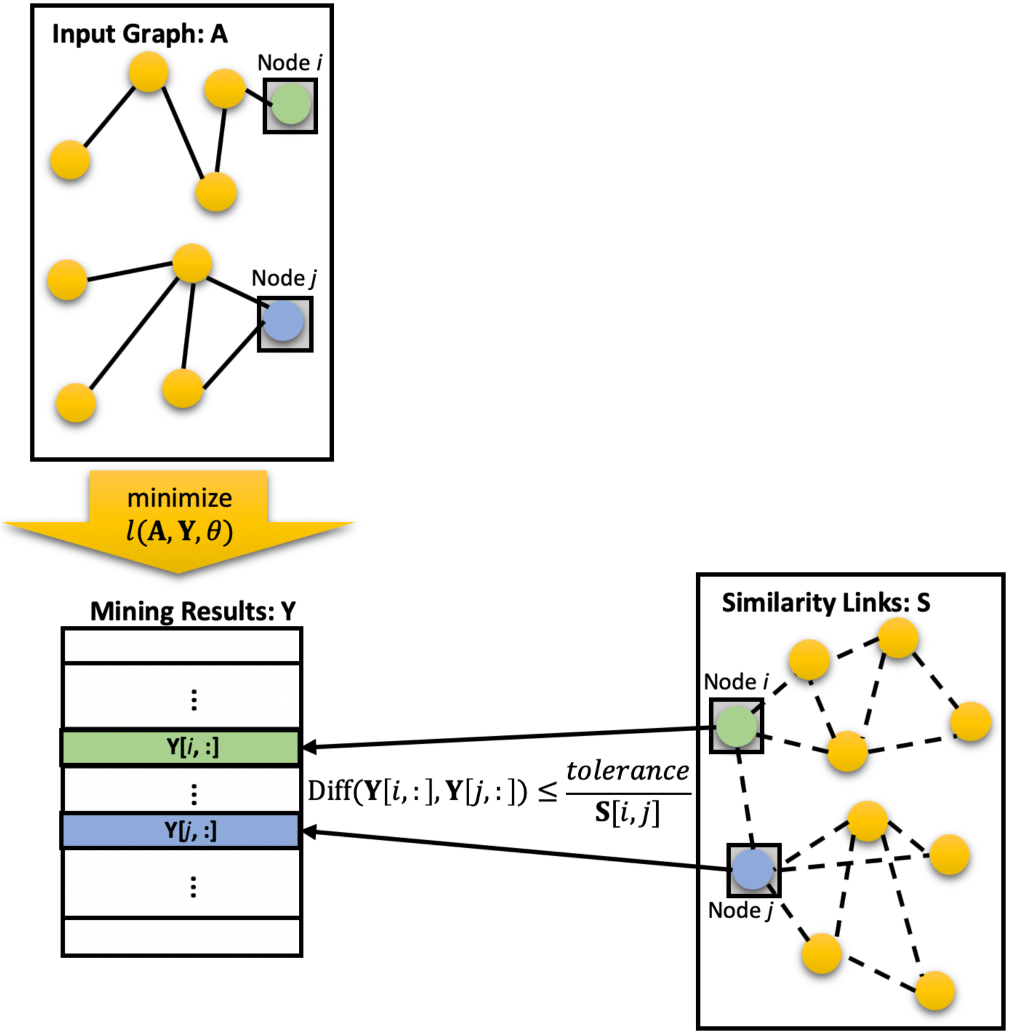
- **Research questions**
  - **RQ1. Measure:** how to quantitatively measure individual bias?
    - **Problem #1:** InFoRM measure problem
  - **RQ2. Algorithms:** how to enforce individual fairness?
    - **Problem #2:** InFoRM algorithms problem
  - **RQ3. Cost:** what is the cost of individual fairness?
    - **Problem #3:** InFoRM cost problem



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Problem #1: InFoRM Measure

- **Questions**
  - How to determine if the mining results are fair?
  - How to quantitatively measure the overall bias?
- **Input**
  - Node-node similarity matrix $\mathbf{S}$
    - Non-negative, symmetric
  - Graph mining algorithm $l(\mathbf{A}, \mathbf{Y}, \theta)$
    - Loss function $l(\cdot)$
    - Additional set of parameters θ
  - Fairness tolerance parameter ϵ
- **Output**
  - Binary decision on whether the mining result is fair
  - Individual bias measure $\text{Bias}(\mathbf{Y}, \mathbf{S})$



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

InFoRM Measure: Formulation

- **Principle:** similar nodes → similar mining results
- **Mathematical formulation**

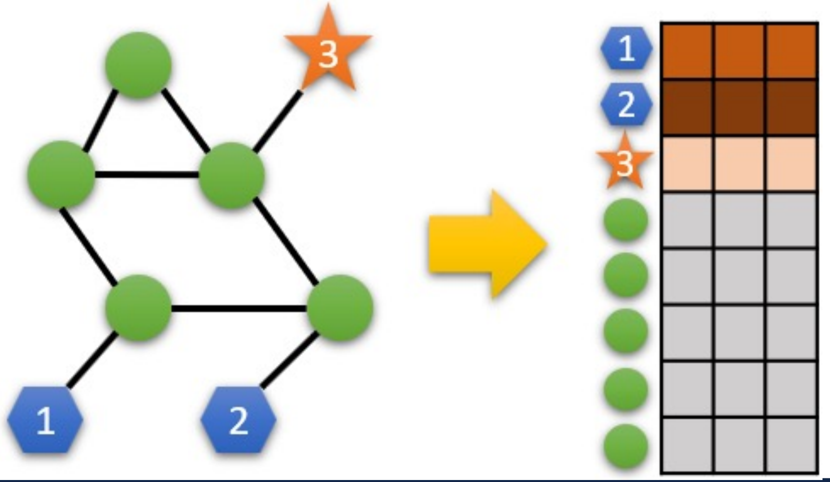
$$\|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \leq \frac{\epsilon}{\mathbf{S}[i,j]} \quad \forall i,j = 1,\dots,n$$

  – **Intuition:** if $\mathbf{S}[i,j]$ is high, $\frac{\epsilon}{\mathbf{S}[i,j]}$ is small → push $\mathbf{Y}[i,:]$ and $\mathbf{Y}[j,:]$ to be more similar

  – **Observation:** inequality should hold for *every* pairs of nodes i and j

    - **Limitation:** too many constraints → too restrictive to be fulfilled

- **Relaxed criteria:** $\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \mathbf{S}[i,j] = 2\mathrm{Tr}(\mathbf{Y}'\mathbf{L_S}\mathbf{Y}) \leq m\epsilon = \delta$



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.
[2] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R.. Fairness through Awareness. ITCS 2012.

InFoRM Measure: Solution

- **InFoRM (<u>In</u>dividual <u>F</u>airness <u>o</u>n G<u>R</u>aph <u>M</u>ining)**
  - **Given:** (1) a graph mining result $\mathbf{Y}$; (2) a symmetric similarity matrix $\mathbf{S}$; and (3) a constant fairness tolerance δ
  - $\mathbf{Y}$ is individually fair w.r.t. $\mathbf{S}$ if it satisfies

$$\mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y}) \leq \frac{\delta}{2}$$

  - Overall individual bias is $\mathrm{Bias}(\mathbf{Y}, \mathbf{S}) = \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

InFoRM Measure: Lipschitz Property

- $(\boldsymbol{D_1}, \boldsymbol{D_2})$**-Lipschitz property:** a function f is (D_1, D_2)-Lipschitz if it satisfies

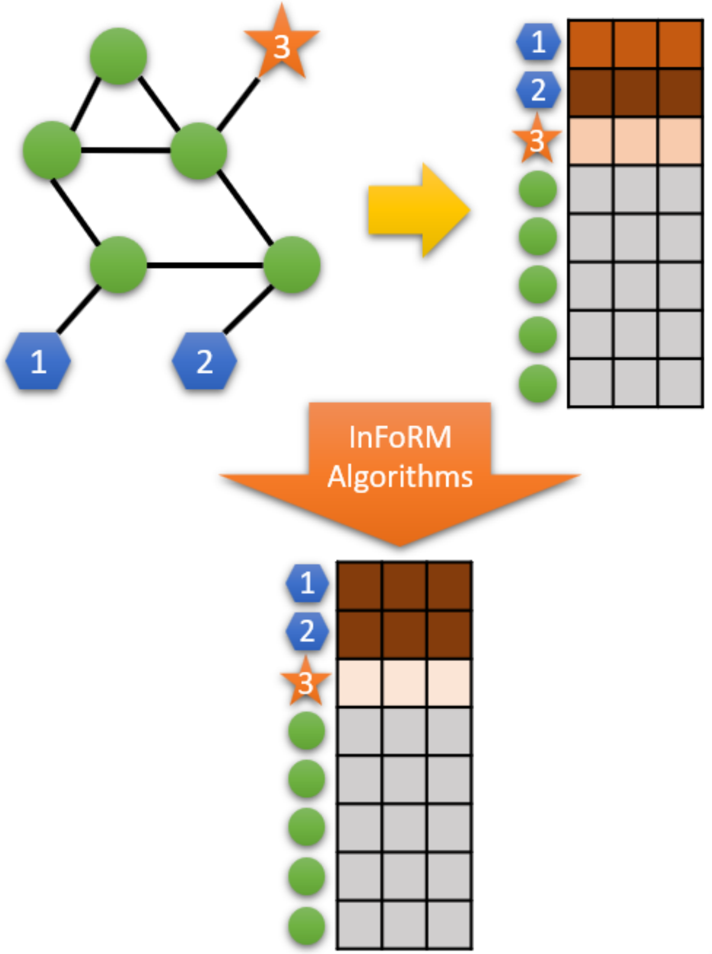
$$D_1\big(f(i), f(j)\big) \leq L D_2(i, j), \forall (x, y)$$

  - L is Lipschitz constant
- InFoRM naturally satisfies (D_1, D_2)-Lipschitz property as long as
  - $f(i) = \mathbf{Y}[i, :]$
  - $D_1\big(f(i), f(j)\big) = \|\mathbf{Y}[i, :] - \mathbf{Y}[j, :]\|_F^2$, $D_2(i, j) = \frac{1}{\mathbf{S}[i,j]}$
- Lipschitz constant of InFoRM is ϵ

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.
[2] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R.. Fairness through Awareness. ITCS 2012.

Problem #2: InFoRM Algorithms

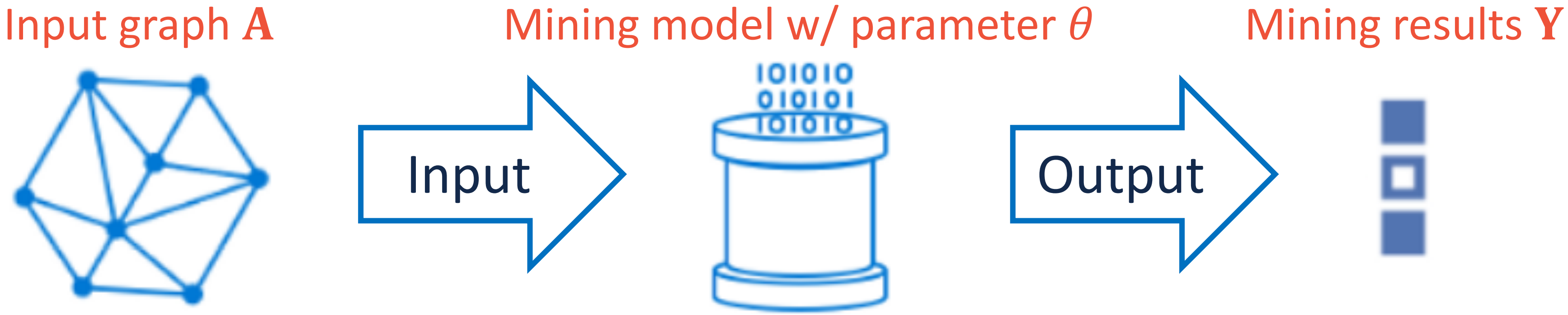
- **Question:** how to mitigate the bias of the mining results?
- **Input**
  - – Node-node similarity matrix $\mathbf{S}$
  - – Graph mining algorithm $l(\mathbf{A}, \mathbf{Y}, \theta)$
  - – Individual bias measure $\mathrm{Bias}(\mathbf{Y}, \mathbf{S})$
    - Defined in the previous problem (InFoRM Measures)
- **Output:** revised mining result $\mathbf{Y}^*$ that minimizes
  - – Task-specific loss function $l(\mathbf{A}, \mathbf{Y}, \theta)$
  - – Individual bias measure $\mathrm{Bias}(\mathbf{Y}, \mathbf{S})$



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Individual Bias Mitigation

- **Graph mining pipeline**



- **Observation:** bias can be introduced/amplified in each component
  - **Solution:** bias can be mitigated in each part
- **Algorithmic frameworks**
  - Debiasing the input graph
  - Debiasing the mining model
  - Debiasing the mining results

mutually complementary

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Debiasing the Input Graph

- **Goal:** bias mitigation via a pre-processing strategy
- **Intuition:** learn a new topology of graph $\tilde{\mathbf{A}}$ such that
  - $\tilde{\mathbf{A}}$ is as similar to the original graph $\mathbf{A}$ as possible
  - Bias of mining results on $\tilde{\mathbf{A}}$ is minimized
- **Optimization problem**

Consistency in graph topology

$$\min_{\mathbf{Y}} \quad J = \left\|\tilde{\mathbf{A}} - \mathbf{A}\right\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$$
$$\text{s.t.} \quad \mathbf{Y} = \mathrm{argmin}_{\mathbf{Y}} \; l(\tilde{\mathbf{A}}, \mathbf{Y}, \theta)$$

Bias measure

- **Challenge:** bi-level optimization
  - **Solution:** exploration of KKT conditions

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.
[2] Mei, S., & Zhu, X.. Using Machine Teaching to Identify Optimal Training-set Attacks on Machine Learners. AAAI 2015.

Problem Reduction

- Considering the KKT conditions,

$$\begin{aligned} \min_{\mathbf{Y}} \quad & J = \left\|\tilde{\mathbf{A}} - \mathbf{A}\right\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y}) \\ \mathrm{s.t.} \quad & \partial_{\mathbf{Y}} l\left(\tilde{\mathbf{A}}, \mathbf{Y}, \theta\right) = 0 \end{aligned}$$

- **Proposed method**
  - (1) Fix $\tilde{\mathbf{A}}$ ($\tilde{\mathbf{A}} = \mathbf{A}$ at initialization), find $\mathbf{Y}$ using current $\tilde{\mathbf{A}}$
  - (2) Fix $\mathbf{Y}$, update $\tilde{\mathbf{A}}$ by gradient descent
  - (3) Iterate between (1) and (2)
- **Problem:** how to compute the gradient w.r.t. $\tilde{\mathbf{A}}$?

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Gradient Computation

Key component to calculate

- **Computing gradient w.r.t. $\widetilde{\mathbf{A}}$**

$$\frac{\partial J}{\partial \widetilde{\mathbf{A}}} = 2(\widetilde{\mathbf{A}} - \mathbf{A}) + \alpha \boxed{\left[\mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L_S}\frac{\partial \tilde{\mathbf{Y}}}{\partial \widetilde{\mathbf{A}}[i,j]}\right)\right]}$$

$$\frac{\mathrm{d}J}{\mathrm{d}\widetilde{\mathbf{A}}} = \begin{cases} \frac{\partial J}{\partial \widetilde{\mathbf{A}}} + \left(\frac{\partial J}{\partial \widetilde{\mathbf{A}}}\right)^T - \mathrm{diag}\left(\frac{\partial J}{\partial \widetilde{\mathbf{A}}}\right), & \text{if undirected} \\ \frac{\partial J}{\partial \widetilde{\mathbf{A}}}, & \text{if directed} \end{cases}$$

  – $\tilde{\mathbf{Y}}$ satisfies $\partial_{\mathbf{Y}} l(\widetilde{\mathbf{A}}, \mathbf{Y}, \theta) = 0$
  – $\mathbf{H} = \left[\mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L_S}\frac{\partial \tilde{\mathbf{Y}}}{\partial \widetilde{\mathbf{A}}[i,j]}\right)\right]$ is a matrix with $\mathbf{H}[i,j] = \mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L_S}\frac{\partial \tilde{\mathbf{Y}}}{\partial \widetilde{\mathbf{A}}[i,j]}\right)$

- **Question:** How to efficiently calculate $\mathbf{H}$?

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Instantiation #1: PageRank

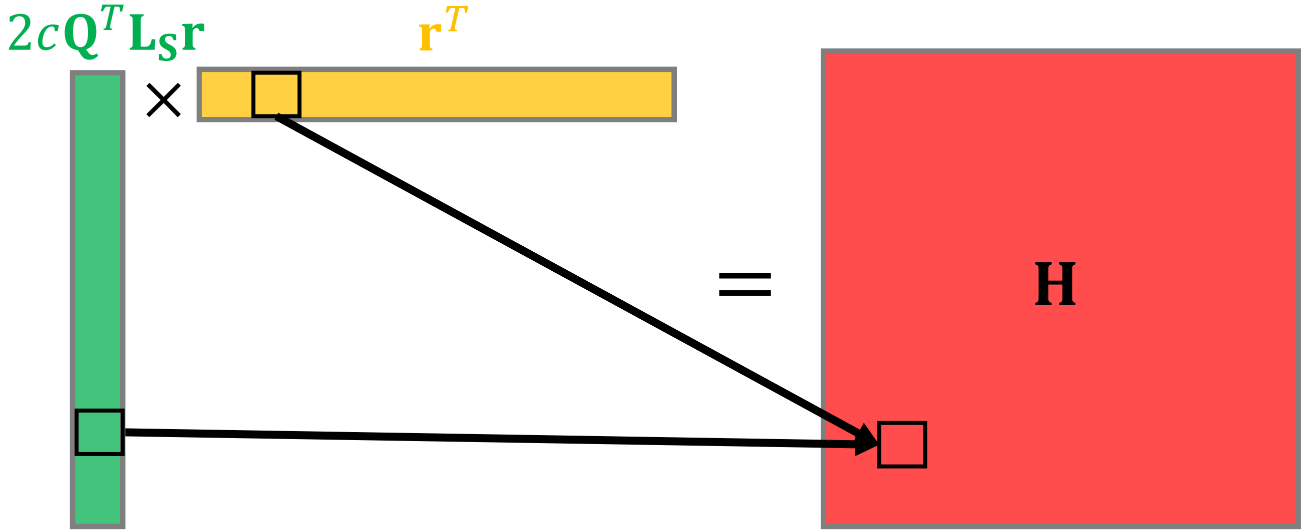
- **Goal:** efficient calculation of $\mathbf{H}$ for PageRank
- **Mining results**

$$\mathbf{r} = (1 - c)\mathbf{Q}\mathbf{e}$$

- **Partial derivatives**

$$\mathbf{H} = 2c\mathbf{Q}^T\mathbf{L}_\mathbf{S}\mathbf{r}\mathbf{r}^T$$

  - $\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$
- **Time complexity**
  - Straightforward: $O(n^3)$
  - Ours: $O(m_1 + m_2 + n)$
    - $m_\mathbf{A}$: number of edges in $\mathbf{A}$
    - $m_\mathbf{S}$: number of edges in $\mathbf{S}$
    - n: number of nodes



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Instantiation #2: Spectral Clustering

- **Goal:** efficient calculation of $\mathbf{H}$ for spectral clustering
- **Mining results**

$$\mathbf{U} = \text{eigenvectors with } k \text{ smallest eigenvalues}$$

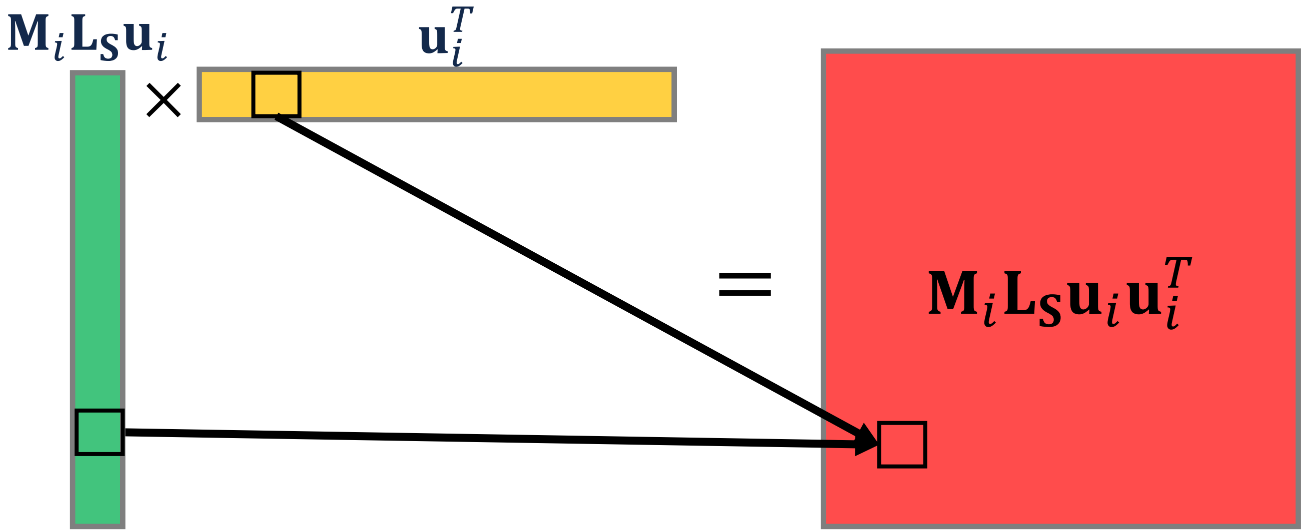
- **Partial derivatives**

Vectorize $\mathrm{diag}(\mathbf{M}_i\mathbf{L}_\mathbf{S}\mathbf{u}_i\mathbf{u}_i')$ and stack it n times

Low-rank

$$\mathbf{H} = 2\sum_{i=1}^{k}\left(\boxed{\mathrm{diag}\left(\mathbf{M}_i\mathbf{L}_\mathbf{S}\mathbf{u}_i\mathbf{u}_i^T\right)\mathbf{1}_{n\times n}} - \boxed{\mathbf{M}_i\mathbf{L}_\mathbf{S}\mathbf{u}_i\mathbf{u}_i^T}\right)$$

  - $(\lambda_i, \mathbf{u}_i) = i$-th smallest eigenpair
  - $\mathbf{M}_i = (\lambda_i\mathbf{I} - \mathbf{L}_\mathbf{A})^+$
- **Time complexity**
  - Straightforward: $O(k^2(m+n) + k^3 n + kn^3)$
  - Ours: $O\big((k+r)(m_1+n) + k(m_2+n) + (k+r)^2 n\big)$
    - k: number of clusters
    - r: number of largest eigenvalues
    - m_1: number of edges in $\mathbf{A}$
    - m_2: number of edges in $\mathbf{S}$
    - n: number of nodes



[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Instantiation #3: LINE (1st)

- **Goal:** efficient calculation of $\mathbf{H}$ for LINE (1st)
- **Mining results**

$$\mathbf{Y}[i,:]\mathbf{Y}[j,:]^T = \log \frac{T(\widetilde{\mathbf{A}}[i,j] + \widetilde{\mathbf{A}}[j,i])}{d_i d_j^{3/4} + d_i^{3/4} d_j} - \log b$$

  - d_i = outdegree of node i, $T = \sum_{i=1}^{n} d_i^{3/4}$ and b = number of negative samples
- **Partial derivatives**

Element-wise in-place calculation

Vectorize $\mathrm{diag}(\mathbf{B}\mathbf{L}_\mathbf{S})$ and stack it n times

$$\mathbf{H} = \boxed{2f\left(\widetilde{\mathbf{A}} + \widetilde{\mathbf{A}}^T\right) \circ \mathbf{L}_\mathbf{S}} - \boxed{2\mathrm{diag}(\mathbf{B}\mathbf{L}_\mathbf{S})\mathbf{1}_{n\times n}}$$

  - $f(\cdot)$ calculates Hadamard inverse, $\circ$ calculates Hadamard product
  - $\mathbf{B} = \frac{3}{4} f\left(\mathbf{d}^{5/4}\left(\mathbf{d}^{-1/4}\right)^T + \boxed{\mathbf{d}\mathbf{1}_{1\times n}}\right) + f\left(\mathbf{d}^{3/4}\left(\mathbf{d}^{1/4}\right)^T + \boxed{\mathbf{d}\mathbf{1}_{1\times n}}\right)$ with $\mathbf{d}^x[i] = d_i^x$

Stack $\mathbf{d}$ n times

- **Time complexity**
  - Straightforward: $O(n^3)$
  - Ours: $O(m_1 + m_2 + n)$
    - m_1: number of edges in $\mathbf{A}$
    - m_2: number of edges in $\mathbf{S}$
    - n: number of nodes

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Debiasing the Mining Model

- **Goal:** bias mitigation during model optimization
- **Intuition:** optimizing a regularized objective such that
  - Task-specific loss function is minimized
  - Bias of mining results as regularization penalty is minimized
- **Optimization problem**

Task-specific loss function

$$\min_{\mathbf{Y}} \quad J = l(\mathbf{A}, \mathbf{Y}, \theta) + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L_S} \mathbf{Y})$$

Bias measure

- **Solution**
  - **General:** (stochastic) gradient descent $\frac{\partial J}{\partial \mathbf{Y}} = \frac{\partial l(\mathbf{A},\mathbf{Y},\theta)}{\partial \mathbf{Y}} + 2\alpha \mathbf{L_S} \mathbf{Y}$
  - **Task-specific:** specific algorithm designed for the graph mining problem
- **Advantage**
  - Linear time complexity incurred in computing the gradient

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Instantiations: Debiasing the Mining Model

- **PageRank**
  - **Objective function:** $\min_{\mathbf{r}} c\mathbf{r}^T(\mathbf{I}-\mathbf{A})\mathbf{r} + (1-c)\|\mathbf{r}-\mathbf{e}\|_F^2 + \alpha\mathbf{r}^T\mathbf{L_S}\mathbf{r}$
  - **Solution:** $\mathbf{r}^* = c\left(\mathbf{A} - \frac{\alpha}{c}\mathbf{L_S}\right)\mathbf{r}^* + (1-c)\mathbf{e}$
    - PageRank on new transition matrix $\mathbf{A} - \frac{\alpha}{c}\mathbf{L_S}$
    - If $\mathbf{L_S} = \mathbf{I} - \mathbf{S}$, then $\mathbf{r}^* = \left(\frac{c}{1+\alpha}\mathbf{A} + \frac{\alpha}{1+\alpha}\mathbf{S}\right)\mathbf{r}^* + \frac{1-c}{1+\alpha}\mathbf{e}$
- **Spectral clustering**
  - **Objective function:** $\min_{\mathbf{U}} \mathrm{Tr}(\mathbf{U}^T\mathbf{L_A}\mathbf{U}) + \alpha\mathrm{Tr}(\mathbf{U}^T\mathbf{L_S}\mathbf{U}) = \mathrm{Tr}(\mathbf{U}^T\mathbf{L}_{\mathbf{A}+\alpha\mathbf{S}}\mathbf{U})$
  - **Solution:** $\mathbf{U}^*$ = eigenvectors of $\mathbf{L}_{\mathbf{A}+\alpha\mathbf{S}}$ with k smallest eigenvalues
    - Spectral clustering on an augmented graph $\mathbf{A} + \alpha\mathbf{S}$
- **LINE (1st)**
  - **Objective function**

$$\max_{\mathbf{x}_i,\mathbf{x}_j} \log g(\mathbf{x}_j\mathbf{x}_i^T) + b\mathbb{E}_{j'\in P_n}\left[\log g\left(-\mathbf{x}_{j'}\mathbf{x}_i^T\right)\right] - \alpha\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_F^2\mathbf{S}[i,j] \quad \forall i,j = 1,\dots,n$$

  - **Solution:** stochastic gradient descent

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Debiasing the Mining Results

- **Goal:** bias mitigation via a post-processing strategy
- **Intuition:** no access to either the input graph or the graph mining model
- **Optimization problem**

Consistency of mining results, convex

$$\min_{\mathbf{Y}} \quad J = \|\mathbf{Y} - \bar{\mathbf{Y}}\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}^T \mathbf{L}_{\mathbf{S}} \mathbf{Y})$$

Bias measure, convex

  - $\bar{\mathbf{Y}}$ is the vanilla mining results
- **Solution:** $(\mathbf{I} + \alpha \mathbf{S})\mathbf{Y}^* = \bar{\mathbf{Y}}$
  - Convex loss function as long as $\alpha \geq 0$ → global optima by $\frac{\partial J}{\partial \mathbf{Y}} = 0$
  - Solve by conjugate gradient (or other linear system solvers)
- **Advantages**
  - No knowledge needed on the input graph
  - Model-agnostic

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Problem #3: InFoRM Cost

- **Question:** how to quantitatively characterize the cost of individual fairness?
- **Input**
  - Vanilla mining result $\bar{\mathbf{Y}}$
  - Debiased mining result $\mathbf{Y}^*$
    - Learned by the previous problem (InFoRM Algorithms)
- **Output:** an upper bound of $\|\bar{\mathbf{Y}} - \mathbf{Y}^*\|_F$
- **Debiasing methods**
  - Debiasing the input graph
  - Debiasing the mining model
  - Debiasing the mining results → main focus

depend on specific graph topology/mining model

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

InFoRM Cost: Debiasing the Mining Results

- **Given**
  - A graph with n nodes and adjacency matrix $\boldsymbol{A}$
  - A node-node similarity matrix $\boldsymbol{S}$
  - Vanilla mining results $\bar{\mathbf{Y}}$
  - Debiased mining results $\mathbf{Y}^* = (\mathbf{I} + \alpha\mathbf{S})^{-1}\bar{\mathbf{Y}}$
- If $\|\boldsymbol{S} - \mathbf{A}\|_F = \Delta$, we have

$$\|\bar{\mathbf{Y}} - \mathbf{Y}^*\|_F \le 2\alpha\sqrt{n}\left(\Delta + \sqrt{rank(\mathbf{A})}\sigma_{\max}(\mathbf{A})\right)\|\bar{\mathbf{Y}}\|_F$$

- **Observation:** the cost of debiasing the mining results depends on
  - The number of nodes n (i.e., size of the input graph)
  - The difference Δ between $\mathbf{A}$ and $\mathbf{S}$
  - The rank of $\boldsymbol{A}$
  - The largest singular value of $\boldsymbol{A}$

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

InFoRM: Experiment

- **Graph mining task:** PageRank
- **Observation:** effective in mitigating bias while preserving the performance of the vanilla algorithm with relatively small changes to the original mining results
  - Similar observations for spectral clustering and LINE (1st)

Datasets	Jaccard Index						Cosine Similarity					
Debiasing the Input Graph												
	Diff	KL	Prec@50	NDCG@50	Reduce	Time	Diff	KL	Prec@50	NDCG@50	Reduce	Time
Twitch	0.109	5.37×10^{-4}	1.000	1.000	24.7%	564.9	0.299	5.41×10^{-3}	0.860	0.899	62.9%	649.3
PPI	0.185	1.90×10^{-3}	0.920	0.944	43.4%	584.4	0.328	8.07×10^{-3}	0.780	0.838	68.7%	636.8
Debiasing the Mining Model												
	Diff	KL	Prec@50	NDCG@50	Reduce	Time	Diff	KL	Prec@50	NDCG@50	Reduce	Time
Twitch	0.182	4.97×10^{-3}	0.940	0.958	62.0%	16.18	0.315	1.05×10^{-2}	0.940	0.957	73.9%	12.73
PPI	0.211	4.78×10^{-3}	0.920	0.942	50.8%	10.76	0.280	9.56×10^{-3}	0.900	0.928	67.5%	10.50
Debiasing the Mining Results												
	Diff	KL	Prec@50	NDCG@50	Reduce	Time	Diff	KL	Prec@50	NDCG@50	Reduce	Time
Twitch	0.035	9.75×10^{-4}	0.980	0.986	33.9%	0.033	0.101	5.84×10^{-3}	0.940	0.958	44.6%	0.024
PPI	0.045	1.22×10^{-3}	0.940	0.958	27.0%	0.020	0.112	6.97×10^{-3}	0.940	0.958	45.0%	0.019

[1] Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.

Overview of Part II

Individual Fairness on Graphs

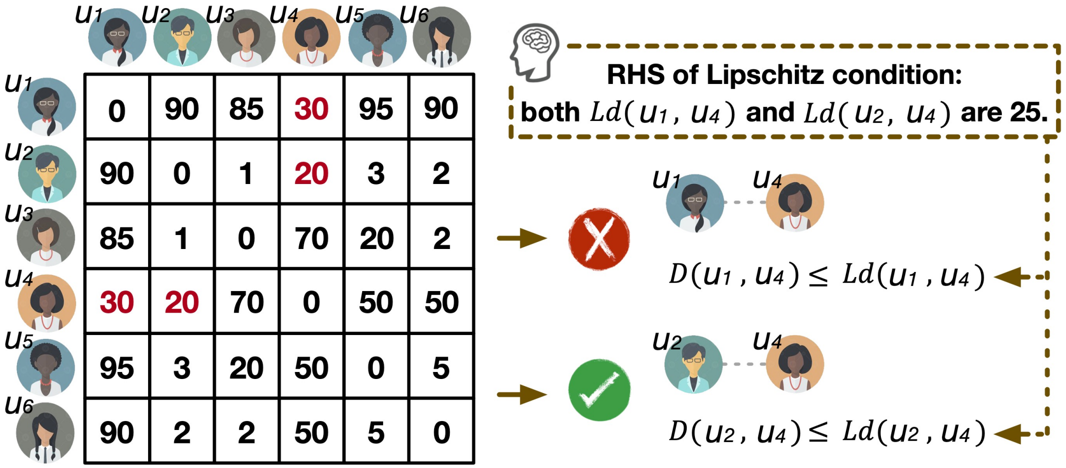
- **Laplacian Regularizer**
  - InFoRM
- **Learning-to-Rank**
  - REDRESS

Individual Fairness on Graph Neural Network

- **Goal:** debias a graph neural network (GNN) to ensure individual fairness
- **Key challenge:** distance calibration
  - Lipschitz condition (used in InFoRM)

$$d_1\big(M(x), M(y)\big) \leq L d_2(x, y)$$

  - Direct distance comparison fails to calibrate the differences between different individuals
- **Example**



	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	90	85	30	95	90
u_2	90	0	1	20	3	2
u_3	85	1	0	70	20	2
u_4	30	20	70	0	50	50
u_5	95	3	20	50	0	5
u_6	90	2	2	50	5	0

(a) Outcome distance matrix from distance metric D

(b) Lipschitz condition judgement based on human knowledge

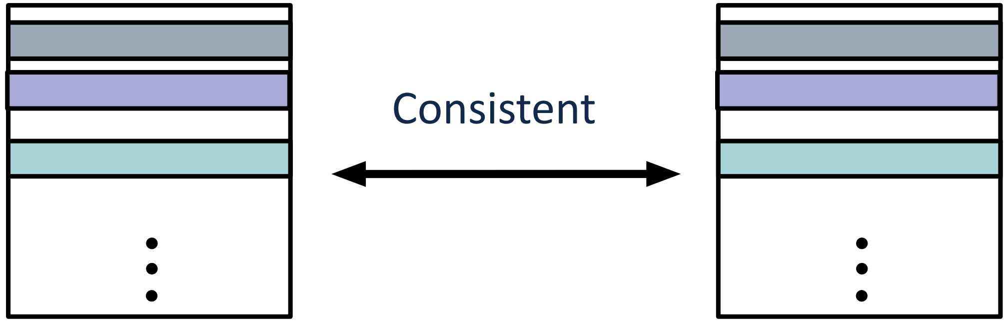
- **Question:** Can we achieve fairness with natural calibration across individuals?

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Ranking basEd InDividual FaiRnESS

- **Ranking-based individual fairness**
  - **Given:** (1) the pairwise node similarity matrix $\mathbf{S}_G$ of the input graph G; (2) the pairwise similarity matrix $\mathbf{S}_{\widehat{\mathbf{Y}}}$ of the GNN output $\widehat{\mathbf{Y}}$
  - $\widehat{\mathbf{Y}}$ is individually fair if, for each node i, it satisfies that

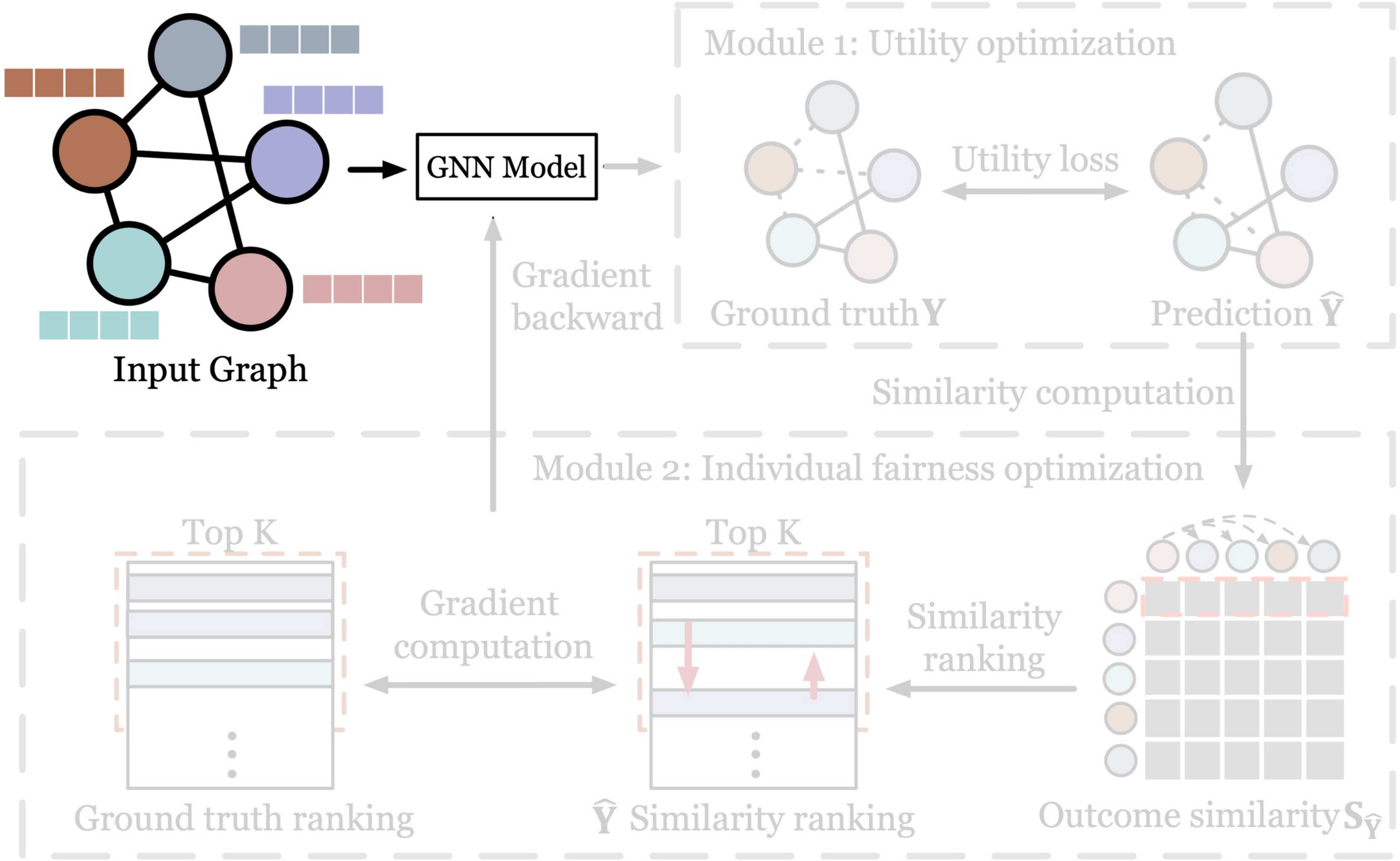
ranking list derived by $\mathbf{S}_G[i,:]$ = ranking list derived by $\mathbf{S}_{\widehat{\mathbf{Y}}}[i,:]$



- **Advantage:** naturally calibrate across individuals
  - No direct distance comparison

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Framework Overview



- **GNN backbone model**
  - Learn node representations
- **Utility maximization**
  - Minimize the downstream task-specific loss
- **Individual fairness optimization**
  - Enforce ranking-based individual fairness

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Backbone Model

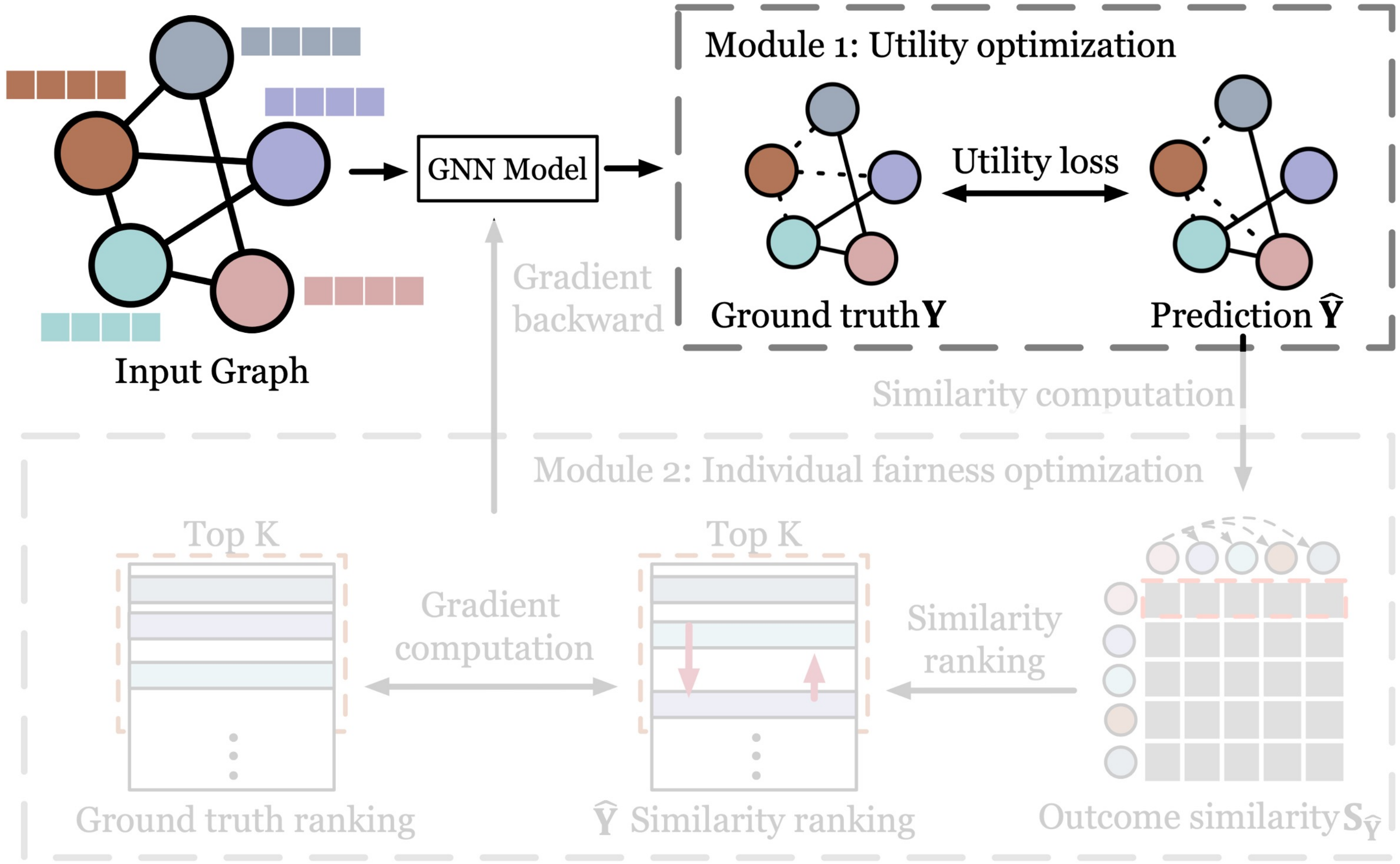
- **Goal:** learn node representations by a GNN
- **Formulation:** l-th GNN Layer

$$h_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \mathrm{AGG}\left(\left\{h_j^{(l)}, \forall j \in \mathcal{N}(i)\right\}\right)\right)$$

  - $h_i^{(l)}$: embedding of node i at l-th layer
  - $\mathbf{W}^{(l)}$: weight parameters at l-th layer
  - $\mathrm{AGG}(\cdot)$: information aggregation function (e.g., mean, weighted sum)
  - $\sigma(\cdot)$: activation function (e.g., ReLU)
  - $\mathcal{N}(i)$: neighborhood set of node i
- **Advantage:** REDRESS works on any GNN model

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Framework Overview



- GNN backbone model
  - Learn node representations
- **Utility maximization**
  - Minimize the downstream task-specific loss
- Individual fairness optimization
  - Enforce ranking-based individual fairness

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Utility Maximization

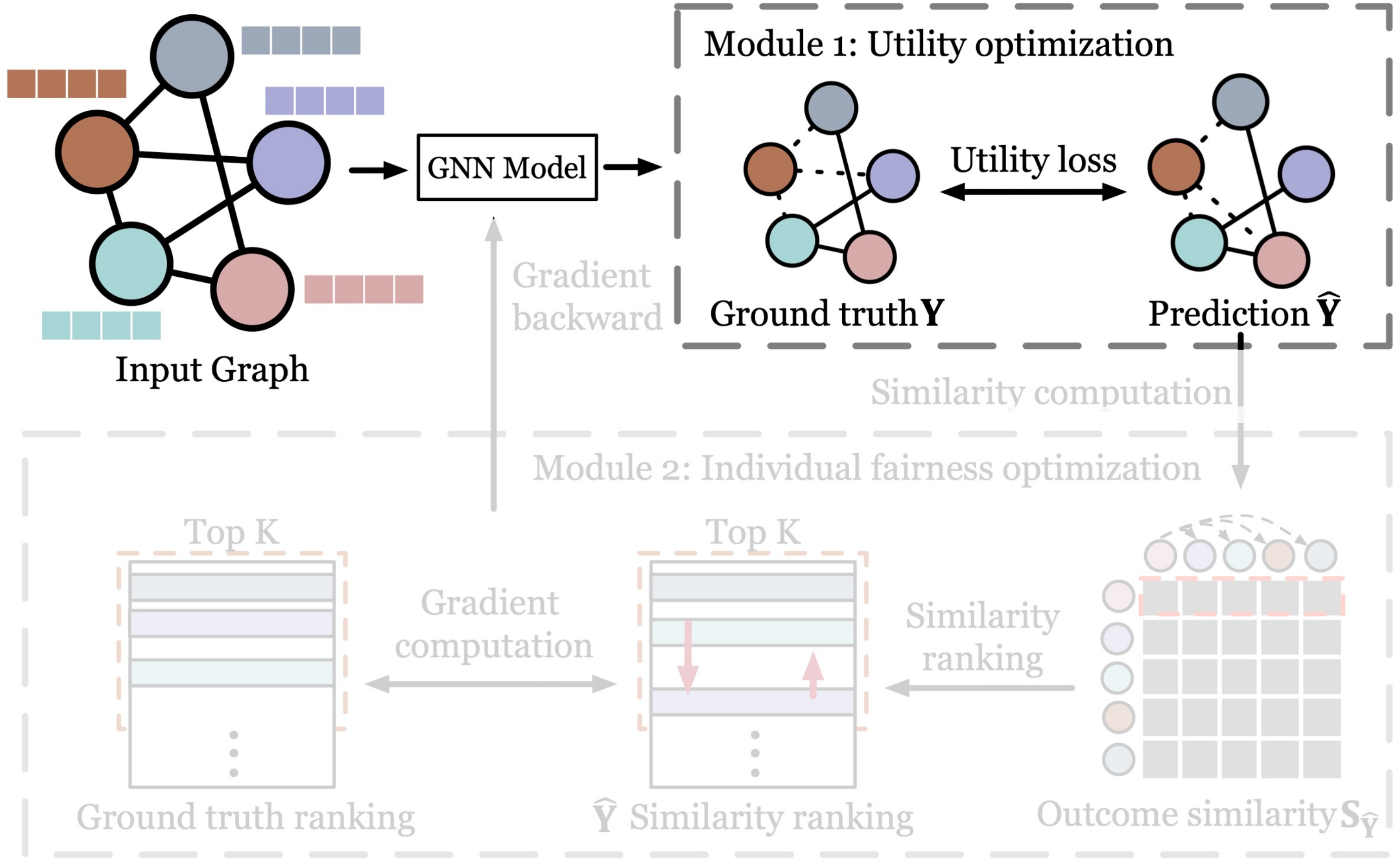
- **Goal:** minimize the downstream task-specific loss function
- **Loss function:** cross-entropy loss

$$L_{\text{utility}} = -\sum_{(i,j)\in\mathcal{T}} \mathbf{Y}[i,j] \log \hat{\mathbf{Y}}[i,j]$$

  - $\mathbf{Y}[i,j]$: i-th row and j-th column in ground truth $\mathbf{Y}$
  - $\hat{\mathbf{Y}}[i,j]$: i-th row and j-th column in GNN predictions $\hat{\mathbf{Y}}$
  - $\mathcal{T}$: a set of tuples
    - **Node classification:** $\mathcal{T}$ is a set of (node, class) tuples
    - **Link prediction:** $\mathcal{T}$ is a set of (node, node) tuples

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Framework Overview

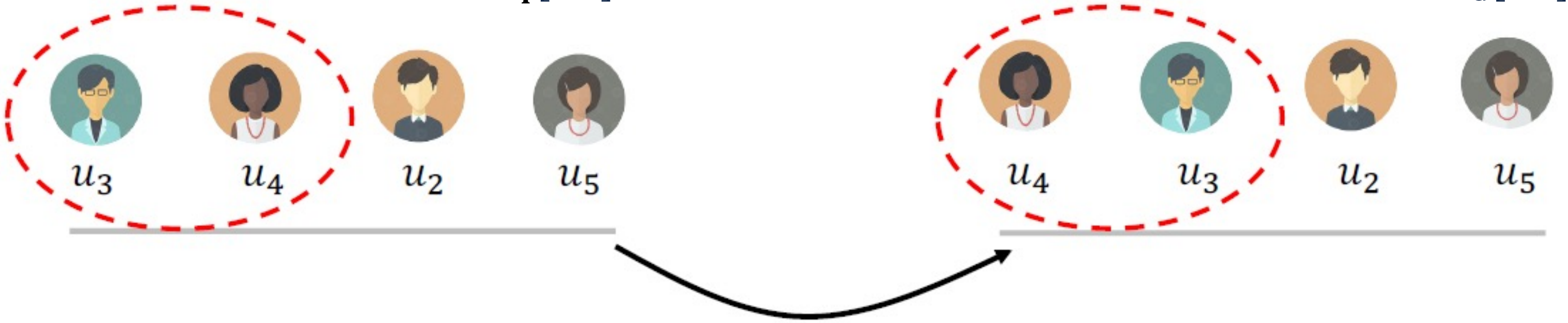


- GNN backbone model
  - Learn node representations
- Utility maximization
  - Minimize the downstream task-specific loss
- **Individual fairness optimization**
  - Enforce ranking-based individual fairness

REDRESS: Individual Fairness Optimization

- **Given:** (1) pairwise node similarity matrix $\mathbf{S}_G$ of input graph G and (2) pairwise similarity matrix $\mathbf{S}_{\hat{\mathbf{Y}}}$ of GNN output $\hat{\mathbf{Y}}$
- **Goal:** for each node i, ensure that the ranking lists derived from $\mathbf{S}_G[i,:]$ and $\mathbf{S}_{\hat{\mathbf{Y}}}[i,:]$ are similar
- **Example:** ranking lists of node u_1

Ranking list derived by $\mathbf{S}_{\hat{\mathbf{Y}}}[1,:]$ **Ranking list derived by $\mathbf{S}_G[1,:]$**



Unfair results
Rankings of u_3 and u_4 are different in two ranking lists

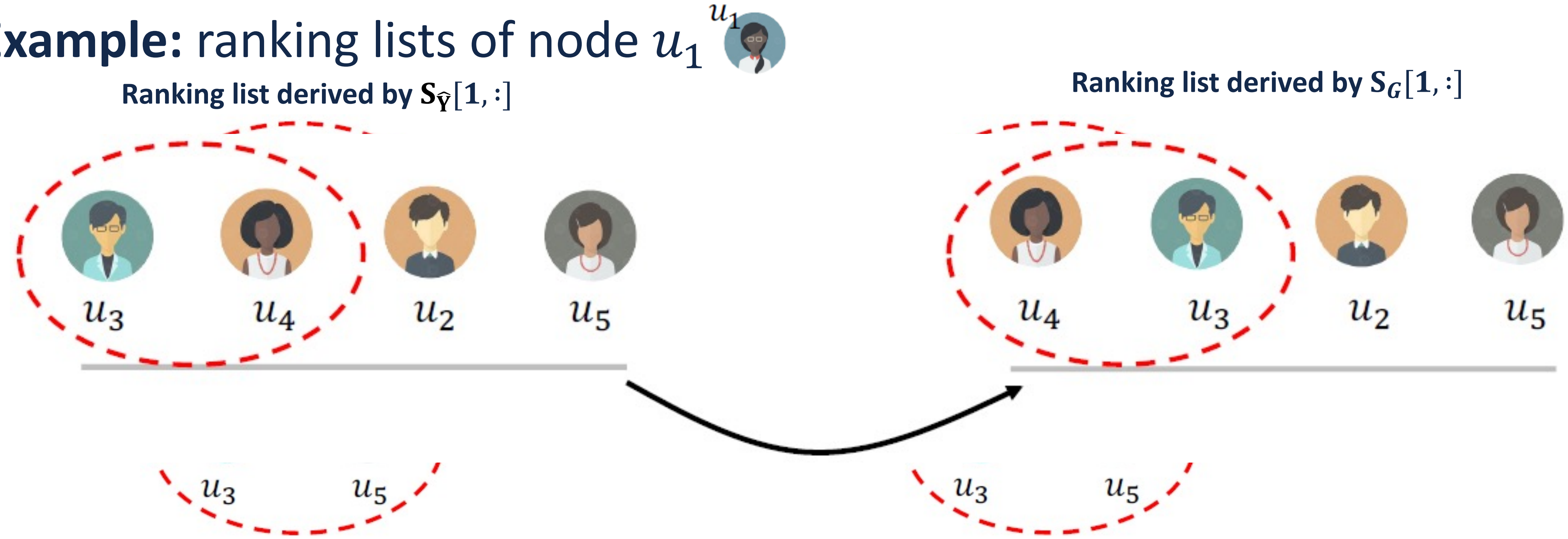
- **Problem:** ranking is a non-differentiable operation

→ loss on the ranking lists will be non-differentiable

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Individual Fairness Optimization

- **Solution**
  - Consider the relative ranking orders of every node pair in $\boldsymbol{S}_G$ and $\mathbf{S}_{\hat{\mathbf{Y}}}$
  - Ensure that every node pair's relative orders are consistent across $\boldsymbol{S}_G$ and $\mathbf{S}_{\hat{\mathbf{Y}}}$
- **Example:** ranking lists of node u_1



[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Relative Ranking Order

- **Key idea:** relative ranking order of u and v = Probability that u ranks higher than v
  - Inspired by learning-to-rank
- **Input space:** pairwise node similarity matrix $\mathbf{S}_G$ of graph G

$$P_{uv}(i) = \frac{1}{2}\left(1 + T_{uv}(i)\right) \qquad T_{uv}(i) = \begin{cases} 1 & u \text{ ranks higher than } v \\ 0 & u \text{ and } v \text{ has the same rank} \\ -1 & v \text{ ranks higher than } u \end{cases}$$

- **Output space:** pairwise similarity matrix $\mathbf{S}_{\hat{\mathbf{Y}}}$ of GNN output $\hat{\mathbf{Y}}$

$$\hat{P}_{uv}(i) = \frac{1}{1 + e^{-\alpha\left(\mathbf{S}_{\hat{\mathbf{Y}}}[i,u] - \mathbf{S}_{\hat{\mathbf{Y}}}[i,v]\right)}}$$

where α is a constant scalar

- **Fairness loss for a node pair**

$$L_{uv}(i) = -P_{uv}(i)\log \hat{P}_{uv}(i) - \left(1 - P_{uv}(i)\right)\log\left(1 - \hat{P}_{uv}(i)\right)$$

[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Scale-up Computation

- **Solution:** focus on top-k similar nodes for each node i in $\mathbf{S}_{\hat{\mathbf{Y}}}$
  - **Individual fairness:** similar outcomes for similar individuals
  - Define $z_{@k}$ = similarity metric for two top-k ranking lists (e.g., NDCG@k)

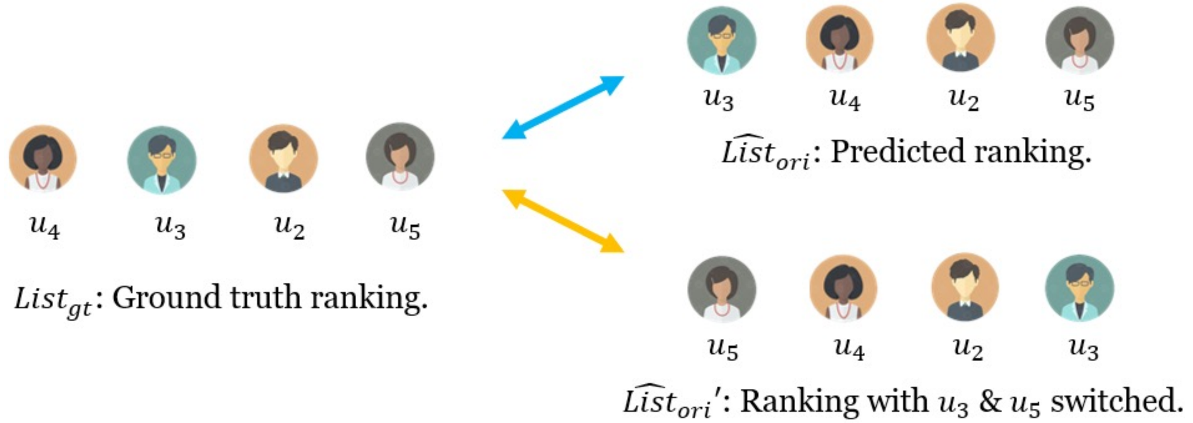
$O(nk^2)$ time complexity

$$L_{\text{fairness}} = \boxed{\sum_i \sum_u \sum_v} L_{uv}(i) |\Delta z_{@k}|_{u,v}$$

  where $|\Delta z_{@k}|_{u,v}$ = absolute value changes in $z_{@k}$ if nodes u and v are swapped
  - High $|\Delta z_{@k}|_{u,v} \rightarrow u$ and v are dissimilar $\rightarrow$ more penalty if ranked wrong

- **Example**

$|\Delta z_{@k}|_{3,5} = |z_{@k}(List_{gt}, \widehat{List}_{ori}) - z_{@k}(List_{gt}, \widehat{List}_{ori}')|$



[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Overall Loss Function

- **Utility loss**

$$L_{\text{utility}} = -\sum_{(i,j)\in\mathcal{T}} \mathbf{Y}[i,j] \log \widehat{\mathbf{Y}}[i,j]$$

- **Fairness loss**

$$L_{uv}(i) = -P_{uv}(i) \log \hat{P}_{uv}(i) - \left(1 - P_{uv}(i)\right) \log\left(1 - \hat{P}_{uv}(i)\right)$$

$$L_{\text{fairness}} = \sum_{i} \sum_{u} \sum_{v} L_{uv}(i) |\Delta z_{@k}|_{u,v}$$

- **Total loss**

$$L = L_{\text{utility}} + \gamma L_{\text{fairness}}$$

where γ is the regularization hyperparameter

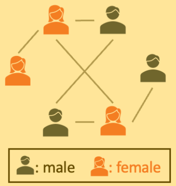
[1] Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021.

REDRESS: Experiment

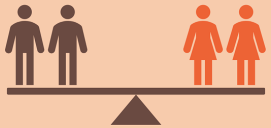
- **Observations for node classification**
  - **Comparable** performance on model utility compared with the best ones
  - **Best** performance on the ranking-based individual fairness
- Similar observations for link prediction

CS	GCN	Vanilla	90.59 ± 0.3 (—)	50.84 ± 1.2 (—)	90.59 ± 0.3 (—)	18.29 ± 0.8 (—)
		InFoRM	88.66 ± 1.1 (−2.13%)	53.38 ± 1.6 (+5.00%)	87.55 ± 0.9 (−3.36%)	19.18 ± 0.9 (+4.87%)
		PFR	87.51 ± 0.7 (−3.40%)	37.12 ± 0.9 (−27.0%)	86.16 ± 0.2 (−4.89%)	11.98 ± 1.3 (−34.5%)
		REDRESS (Ours)	**90.70 ± 0.2 (+0.12%)**	**55.01 ± 1.9 (+8.20%)**	89.16 ± 0.3 (−1.58%)	**21.28 ± 0.3 (+16.4%)**
	SGC	Vanilla	87.48 ± 0.8 (—)	74.00 ± 0.1 (—)	87.48 ± 0.8 (—)	32.36 ± 0.3 (—)
		InFoRM	88.07 ± 0.1 (+0.67%)	74.29 ± 0.1 (+0.39%)	88.65 ± 0.4 (+1.34%)	32.37 ± 0.4 (+0.03%)
		PFR	88.31 ± 0.1 (+0.94%)	48.40 ± 0.1 (−34.6%)	84.34 ± 0.3 (−3.59%)	28.87 ± 0.9 (−10.8%)
		REDRESS (Ours)	**90.01 ± 0.2 (+2.89%)**	**76.60 ± 0.1 (+3.51%)**	**89.35 ± 0.1 (+2.14%)**	**34.24 ± 0.2 (+5.81%)**

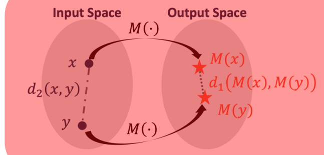
Roadmap



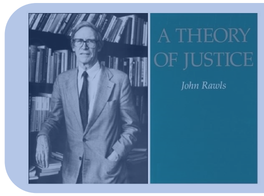
Introduction



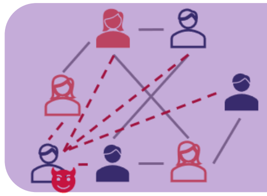
Part I: Group Fairness on Graphs



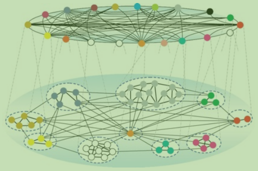
Part II: Individual Fairness on Graphs



Part III: Other Fairness on Graphs



Part IV: Beyond Fairness on Graphs



Part V: Future Trends

Overview of Part III

Other Fairness on Graphs

Counterfactual Fairness
- NIFTY
- GEAR

Degree Fairness
- SL-DSGCN
- RawlsGCN

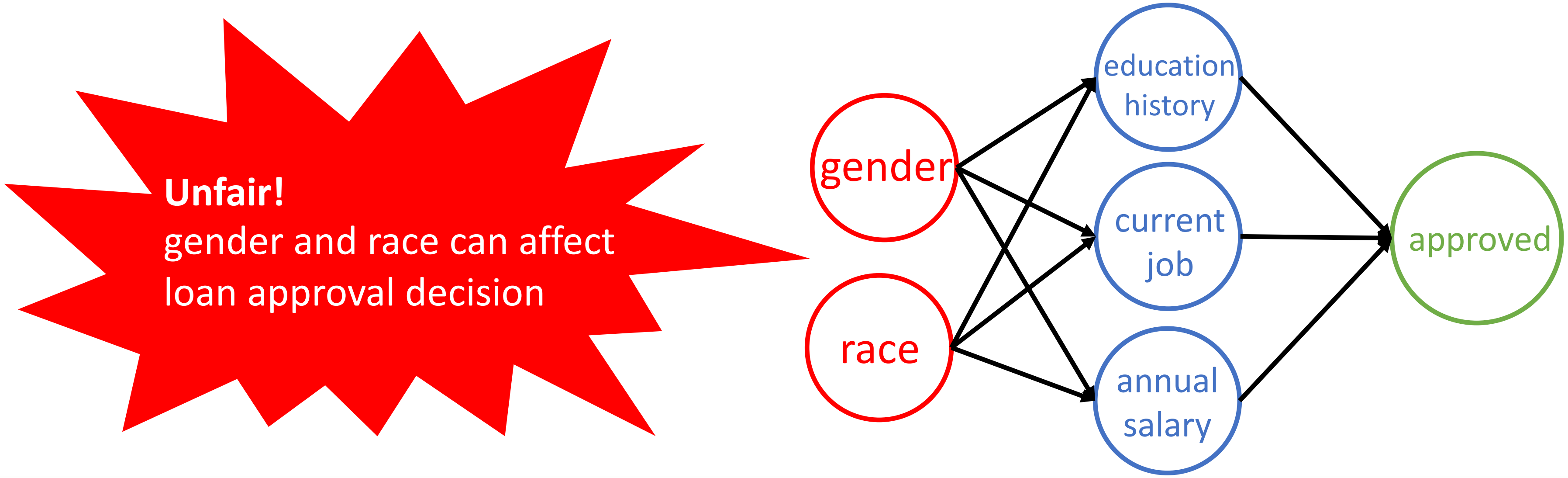
Recap: Counterfactual Fairness

counterfactual version

- **Definition:** same outcomes for 'different versions' of the same candidate

$$\Pr\left(\hat{y}_{s=s_1} = c | s = s_1, x = \mathbf{x}\right) = \Pr\left(\hat{y}_{s=s_2} = c | s = s_2, x = \mathbf{x}\right)$$

  - $\Pr\left(\hat{y}_{s=s_1} = c | s = s_1, x = \mathbf{x}\right)$: version 1 of $\mathbf{x}$ with sensitive demographic s_1
  - $\Pr\left(\hat{y}_{s=s_2} = c | s = s_2, x = \mathbf{x}\right)$: version 2 of $\mathbf{x}$ with sensitive demographic s_2
- **Intuition:** perturbations on the sensitive attribute should not affect the output
- **Example:** causal graph of loan approval



[1] Kusner, M. J., Loftus, J., Russell, C., & Silva, R.. Counterfactual Fairness. NeurIPS 2017.

Preliminary: Stability

- **Definition:** perturbations on the input data should not affect the output too much
- **Mathematical formulation:** Lipschitz condition

$$d_1\big(M(x), M(\tilde{x})\big) \leq L d_2(x, \tilde{x})$$

  - M: a mapping from input to output
  - d_1: distance metric for output
  - d_2: distance metric for input
  - L: Lipschitz constant
  - $\tilde{x}$: perturbed version of original input data x

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

Counterfactual Fairness vs. Stability

- **Given**
  - $\mathbf{A}$: binary adjacency matrix of a graph
  - $\mathbf{x}_u$: feature vector $\mathbf{x}_u$ of a node u
  - $\mathbf{b}_u = [\mathbf{x}_u; \mathbf{A}[u,:]]$: information vector of node u
  - $\tilde{u}$: perturbed version of node u with information vector $\tilde{\mathbf{b}}_u$
    - Perturbation(s) on $\mathbf{x}_u$ or $\mathbf{A}[u,:]$
  - $\tilde{\mathbf{b}}_u$: information vector of node $\tilde{u}$
  - $\tilde{u}^s$: counterfactual version of node u
    - Modification on the value of sensitive attribute s in $\mathbf{x}_u$
  - $\mathrm{ENC}(u)$: an encoder function that learns the embedding of node u
- **Counterfactual fairness**

$$\|\mathrm{ENC}(u) - \mathrm{ENC}(\tilde{u})\|_p = 0$$

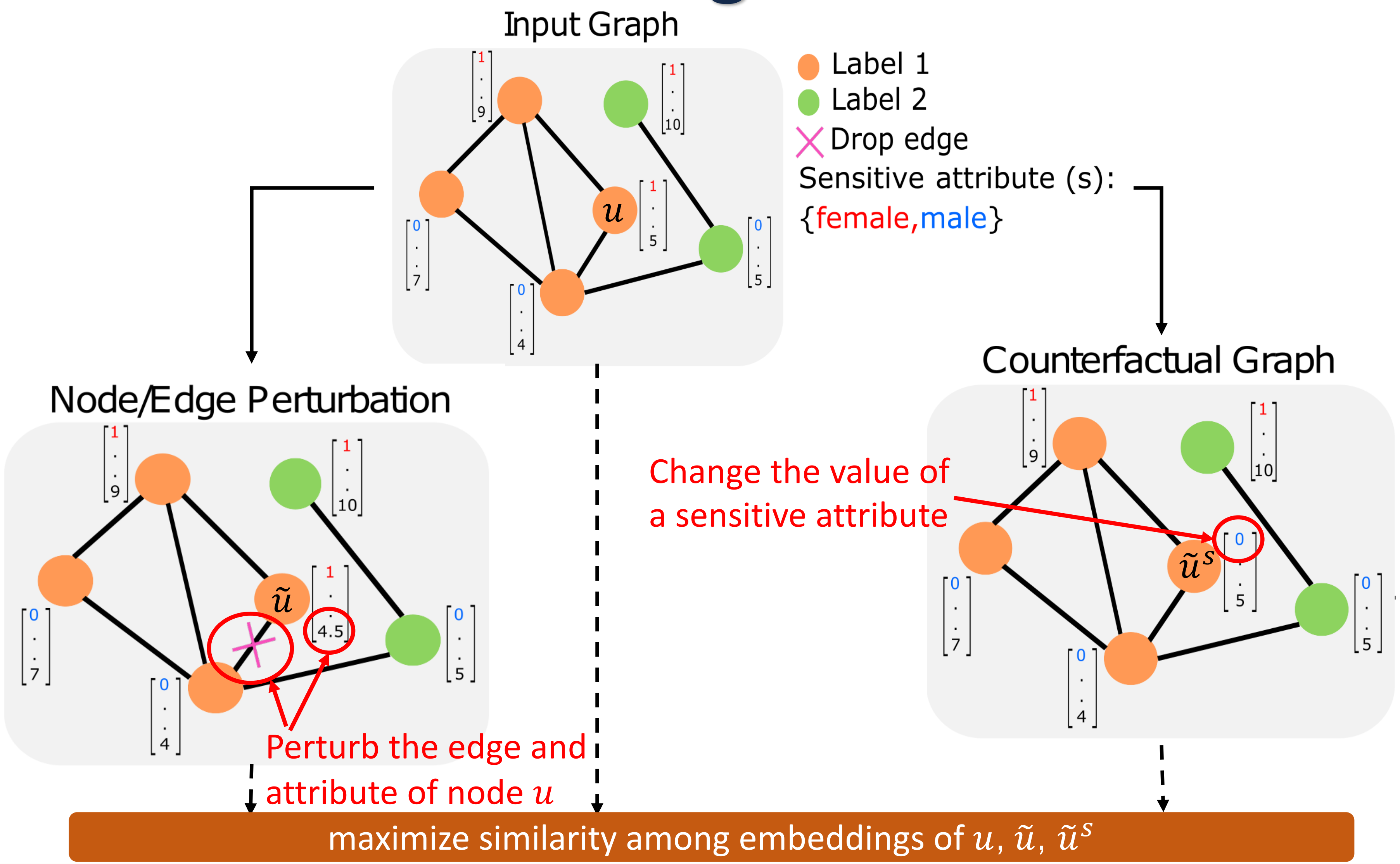
- **Stability**

$$\|\mathrm{ENC}(u) - \mathrm{ENC}(\tilde{u})\|_p \leq L\left\|\tilde{\mathbf{b}}_u - \mathbf{b}_u\right\|_p$$

- **Question:** can we learn node embedding that is both counterfactually fair and stable?

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Contrastive Learning-based Framework



[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Model Architecture

- **Given**
  - $\mathbf{h}_u^{(k)}$: representation of node u at k-th layer
  - $\mathcal{N}(u)$: neighborhood of node u
  - $\mathbf{W}_a^{(k)}$: self-attention weight matrix at k-th layer
  - $\widetilde{\mathbf{W}}_a^{(k)} = \frac{\mathbf{W}_a^{(k)}}{\left\|\mathbf{W}_a^{(k)}\right\|_p}$: Lipschitz-normalization on $\mathbf{W}_a^{(k)}$
    - $\left\|\mathbf{W}_a^{(k)}\right\|_p$: spectral norm of $\mathbf{W}_a^{(k)}$
  - $\mathbf{W}_n^{(k)}$: weight matrix associated with the neighbors of node u
- The k-th NIFTY layer learns node representation by

$$\mathbf{h}_u^{(k)} = \sigma\left(\widetilde{\mathbf{W}}_a^{(k-1)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_n^{(k-1)} \sum_{v\in\mathcal{N}(u)} \mathbf{h}_v^{(k-1)}\right)$$

- NIFTY encoder $\mathrm{ENC}(\cdot) =$ a stack of K NIFTY layers

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Contrastive Loss

- **Goal:** maximize similarity among embeddings of u, $\tilde{u}$, $\tilde{u}^s$
- **Augmented graph:** either (1) edge/attribute perturbed graph or (2) counterfactual graph with modification on the value of sensitive attribute
- **Formulation**

$$L_s(u, \tilde{u}^{\mathrm{aug}}) = \frac{D\left(\mathrm{FC}(\mathbf{z}_u), \mathrm{SG}\left(\mathbf{z}_u^{\mathrm{aug}}\right)\right) + D\left(\mathrm{FC}\left(\mathbf{z}_u^{\mathrm{aug}}\right), \mathrm{SG}(\mathbf{z}_u)\right)}{2}$$

  - $D(\cdot,\cdot)$: cosine distance
  - $\tilde{u}^{\mathrm{aug}}$: counterpart of node u in the augmented graph
  - $\mathbf{z}_u, \mathbf{z}_u^{\mathrm{aug}}$: representation of nodes u and $\tilde{u}^{\mathrm{aug}}$ learned by NIFTY encoder
  - $\mathrm{FC}(\cdot)$: a fully-connected layer for embedding alignment
  - $\mathrm{SG}(\cdot)$: stop-grad operator, stop calculating the gradient with respect to its input

- **Intuition:** minimize L_s $\begin{cases} \mathrm{FC}(\mathbf{z}_u) \text{ and } \mathbf{z}_u^{\mathrm{aug}} \text{ are similar} \\ \mathrm{FC}\left(\mathbf{z}_u^{\mathrm{aug}}\right) \text{ and } \mathbf{z}_u \text{ are similar} \end{cases}$

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Overall Loss Function

- **Overall loss function**

$$L = (1 - \lambda)L_c + \lambda(\mathbb{E}_u[L_s(u, \tilde{u})] + \mathbb{E}_u[L_s(u, \tilde{u}^s)])$$

  – λ: regularization hyperparameter
  – L_c: task-specific loss
    - E.g., cross-entropy loss for node classification
  – $\mathbb{E}_u[L_s(u, \tilde{u})]$: similarity loss of original graph and the edge/attribute perturbed graph
  – $\mathbb{E}_u[L_s(u, \tilde{u}^s)]$: similarity loss of original graph and the counterfactual graph
- **Intuition:** jointly minimize
  – The task-specific loss
  – Distance among embeddings of u, $\tilde{u}$ and $\tilde{u}^s$, for each node u

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Counterfactual Fairness

- **Given**
  - $\mathrm{ENC}(\cdot)$: a K-layer NIFTY encoder
    - $\widetilde{\mathbf{W}}_a^{(k)}$: self-attention weight matrix at k-th layer
  - s: a binary-valued sensitive attribute s
  - u: a node u in the graph
  - $\tilde{u}^s$: the counterfactual version of node u by flipping the value of s
- NIFTY is counterfactually fair with the unfairness upper bounded as follows

$$\|\mathrm{ENC}(u) - \mathrm{ENC}(\tilde{u}^s)\|_p \leq \prod_{k=1}^{K} \left\|\widetilde{\mathbf{W}}_a^{(k)}\right\|_p$$

- **Remarks**
  - Upper bounded counterfactual unfairness (i.e., $\|\mathrm{ENC}(u) - \mathrm{ENC}(\tilde{u}^s)\|_p$)
  - Normalized $\widetilde{\mathbf{W}}_a^{(k)}$ → counterfactually fair $\mathrm{ENC}(u)$

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Stability

- **Given**
  - $\mathrm{ENC}(\cdot)$: a K-layer NIFTY encoder
    - $\widetilde{\mathbf{W}}_a^{(k)}$: self-attention weight matrix at k-th layer
  - s: a binary-valued sensitive attribute
  - $\mathbf{b}_u$: a node u with information vector $\mathbf{b}_u$
  - $\tilde{\mathbf{b}}_u$: perturbed version $\tilde{u}$ of node u with information vector
- NIFTY learns stable node embedding

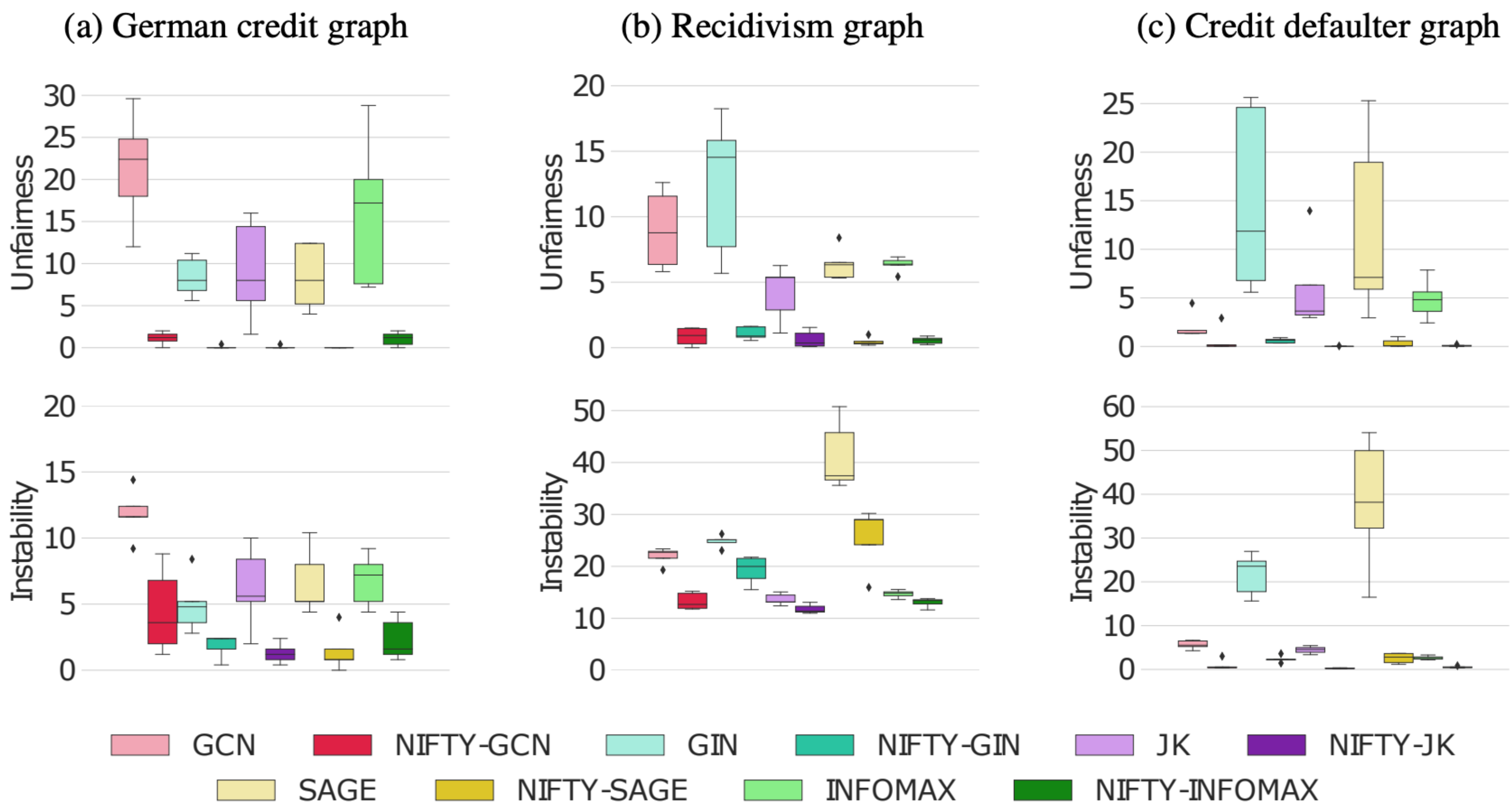
$$\|\mathrm{ENC}(u) - \mathrm{ENC}(\tilde{u})\|_p \leq \prod_{k=1}^{K} \left\|\widetilde{\mathbf{W}}_a^{(k)}\right\|_p \|\mathbf{b}_u - \tilde{\mathbf{b}}_u\|_p$$

- **Remarks**
  - Lipschitz constant = $\prod_{k=1}^{K} \left\|\widetilde{\mathbf{W}}_a^{(k)}\right\|_p$
  - Normalized $\widetilde{\mathbf{W}}_a^{(k)}$ → small Lipschitz constant → stable $\mathrm{ENC}(u)$

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

NIFTY: Experiment

- **Observation:** NIFTY improves both fairness and stability



[1] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

Overview of Part III

Other Fairness on Graphs

Counterfactual Fairness

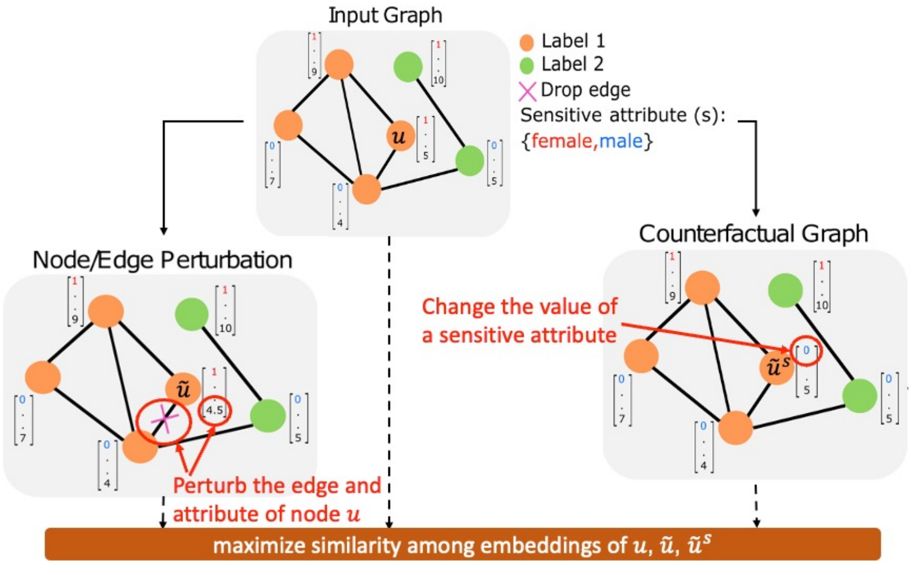
- NIFTY
- GEAR

Degree Fairness

- SL-DSGCN
- RawlsGCN

Limitation: Counterfactual Fairness and NIFTY

- **Counterfactual fairness:** same outcomes for 'different versions' of the same candidate
- **Counterfactual graph generation:** perturbation on the sensitive attribute of central node u
- **Uniqueness of graph data:** change in neighboring nodes could affect the central node
  - Not considered in NIFTY



[1] Kusner, M. J., Loftus, J., Russell, C., & Silva, R.. Counterfactual Fairness. NeurIPS 2017.
[2] Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.

GEAR: Graph Counterfactual Fairness

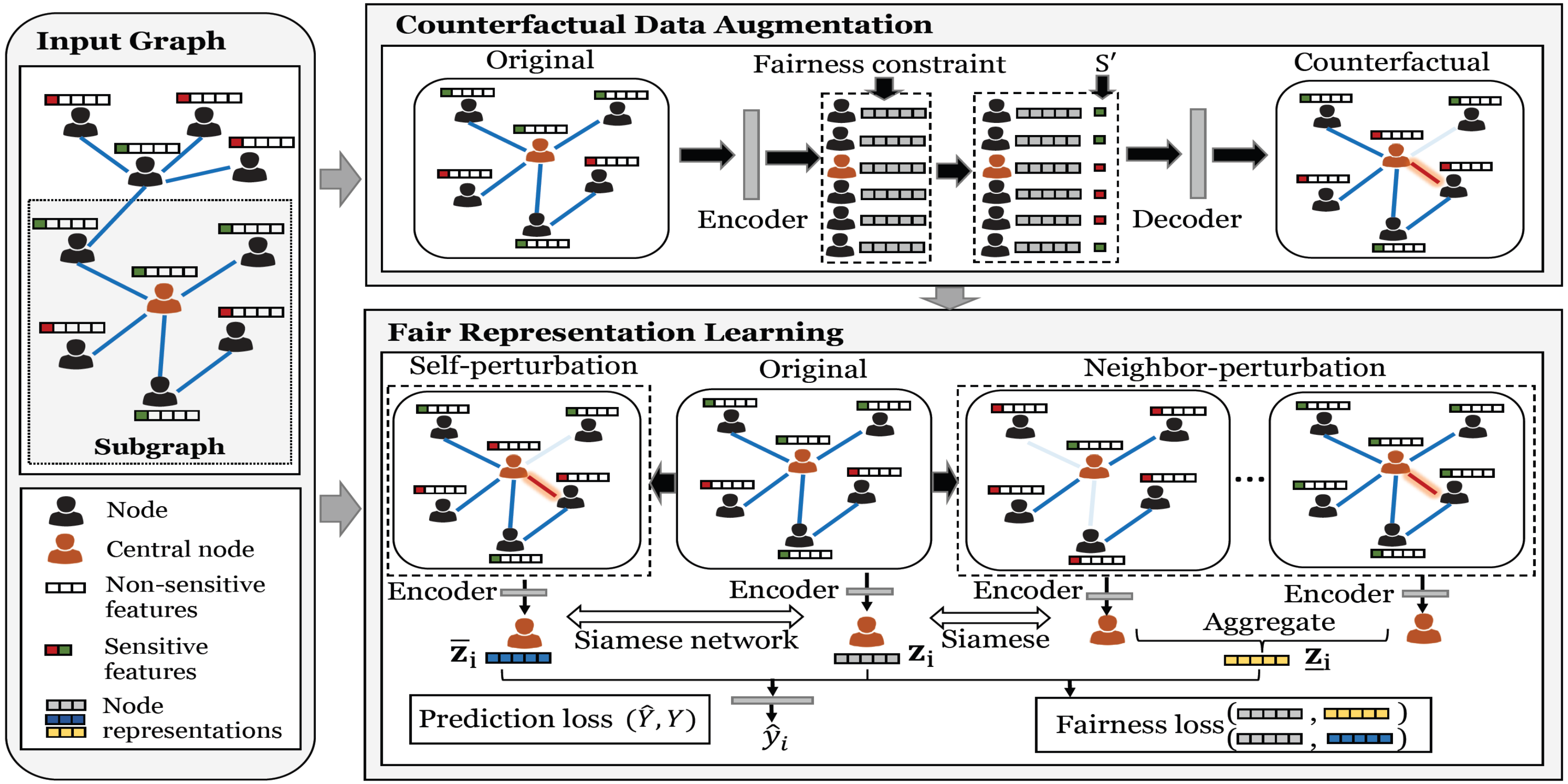
- **Intuition:** same outcomes of a node no matter how the sensitive attribute changes for any node in the graph
- **Given**
  - $G = (\mathbf{A}, \mathbf{X})$: a graph
    - $\mathbf{A}$: adjacency matrix
    - $\mathbf{X}$: node feature matrix
  - $\mathbf{s}$: a vector representing the sensitive attribute of all nodes
    - $\mathbf{s}[i]$ is the sensitive attribute of node i in G
  - $\mathbf{s}'$: the counterfactual version of $\mathbf{s}$ by flipping the sensitive attribute of any node in $\mathbf{A}$
  - $(\mathbf{Y}[i,:])_{S=\mathbf{s},G=(\mathbf{A},\mathbf{X})}$: mining results of node i when the sensitive attribute vector is $\mathbf{s}$ and input graph is $(\mathbf{A}, \mathbf{X})$
- The mining results $\mathbf{Y}$ satisfies graph counterfactual fairness if it satisfies

$$(\mathbf{Y}[i,:])_{S=\mathbf{s},G=(\mathbf{A},\mathbf{X})} = (\mathbf{Y}[i,:])_{S=\mathbf{s}',G=(\mathbf{A},\mathbf{X})}$$

[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

GEAR: Framework Overview

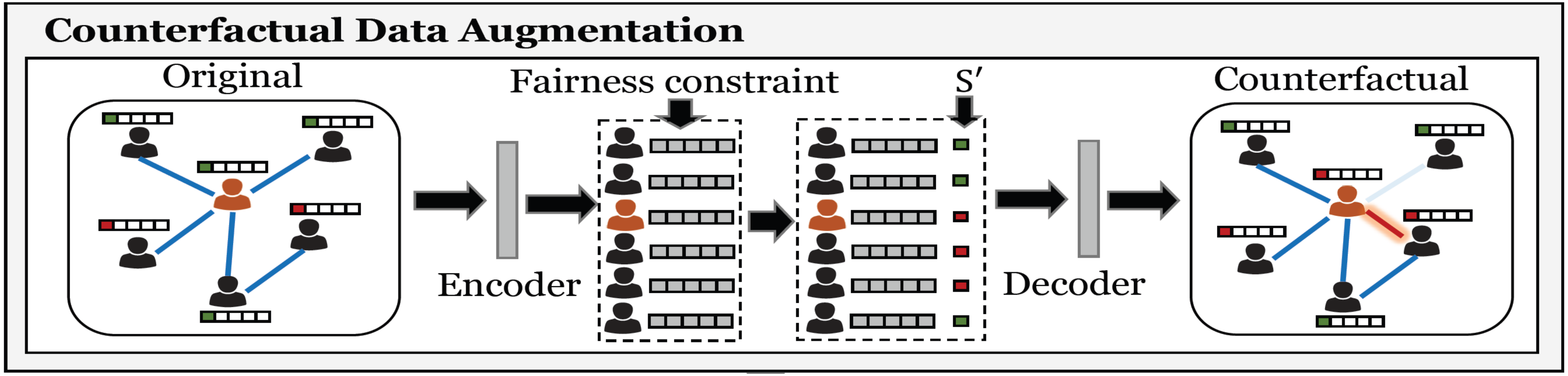
- **Module #1:** counterfactual data augmentation
- **Module #2:** fair representation learning



[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

GEAR: Counterfactual Data Generation

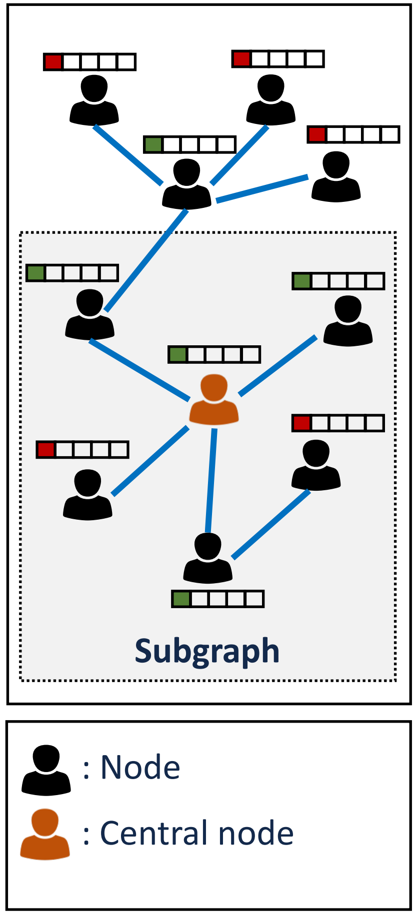
- **Goal:** counterfactual graph generation by perturbing sensitive attribute of arbitrary node(s) in the graph
- **Assumption:** exogenous sensitive attribute → no parent variable in the causal graph
- **Challenges**
  - **C1:** too many possible combinations of perturbation
  - **C2:** modeling of exogenous sensitive attribute



[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

C1: Reducing Number of Counterfactuals

- **Problems:** too many possible combinations of sensitive attribute perturbation
- **Facts**
  - The causal model of a large graph is hard to obtain
  - Each node is mostly influenced by its nearest neighbors
- **Solution:** local subgraph
  - Random walk with restart for proximity computation
  - Top-k node selection for subgraph extraction



[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

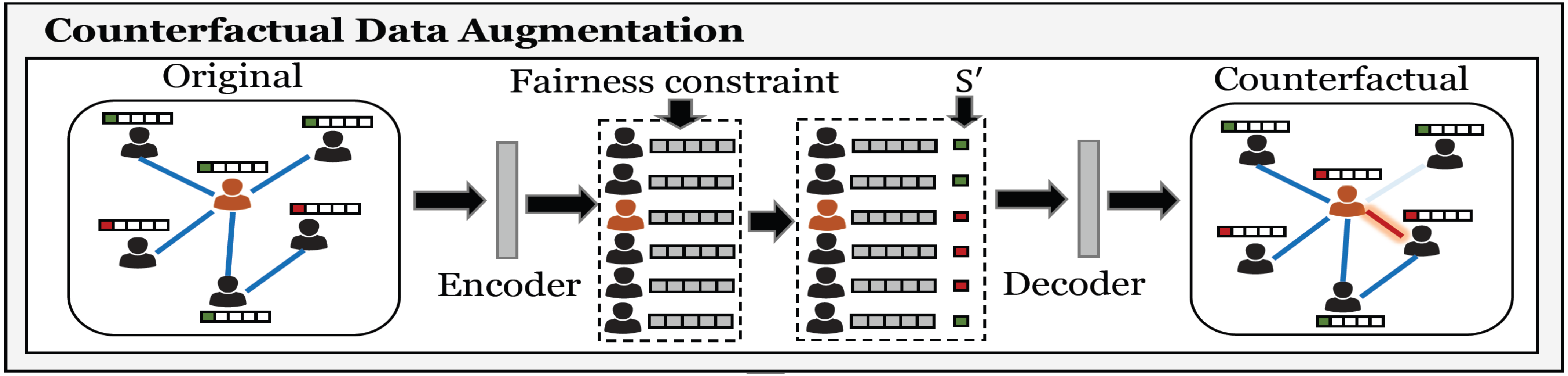
C2: Modeling Exogenous Sensitive Attribute

- **Exogenous sensitive attribute:** no parent variable in the causal graph
  - Cannot be affected by graph structure or any node features
  - Can affect graph structure and other node features
- **Key idea:** decouple the information about graph structure and sensitive attribute
- **Solution:** graph variational auto-encoder (GVAE) + fairness constraints
  - **GVAE:** learn representative embedding about graph structure
  - **Fairness constraints:** decouple the information about graph structure and sensitive attribute
    - **Key idea:** train a discriminator to predict sensitive attribute from embedding
- **Optimization:** alternating stochastic gradient descent
  - Minimize reconstruction loss of GVAE
  - Maximize the prediction error of discriminator

[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

GEAR: Counterfactual Data Generation

- Extract local subgraph of a central node with random walk with restart
- Train a fair GVAE to learn embedding for subgraph reconstruction
- Flip the sensitive attribute of a node in the subgraph
  - **Self-perturbation:** flip the sensitive attribute of the central node
  - **Neighbor perturbation:** flip the sensitive attribute of any nodes except
- Generate two counterfactual subgraphs based on self-perturbation and neighbor perturbation



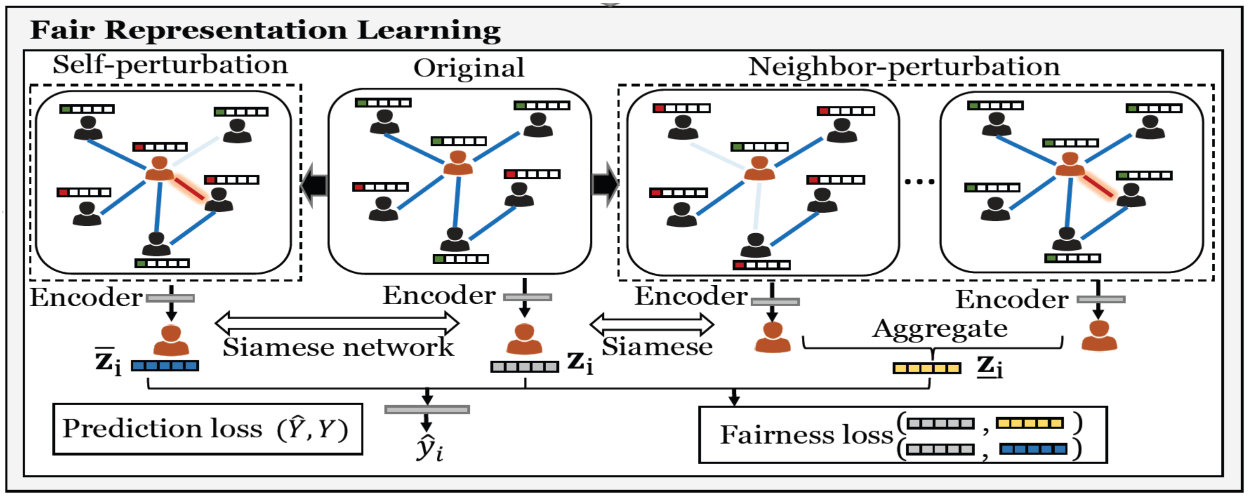
[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

GEAR: Fair Representation Learning

- **Goal:** learn node representation that is invariant to counterfactual graphs
- **Key idea:** for a node u, minimize the distance among
  - Original embedding (z_u),
  - Self-perturbation embedding ($\overline{z_u}$)
  - Neighbor-perturbation embedding ($\underline{z_u}$)
- **Contrastive loss**

$$L = \mathbb{E}_u\left[(1-\lambda_s)d(z_u, \overline{z_u}) + \lambda_s d\left(z_u, \underline{z_u}\right)\right]$$

  - λ_s: hyperparameter
  - d: a distance metric



[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

GEAR: Experiment

- **Observations**
  - GEAR achieves comparable performance in utility metrics and other group fairness metrics
  - GEAR achieves the best performance in graph counterfactual fairness measure

Dataset	Method	Prediction Performance			Fairness			
		Accuracy (↑)	F1-score (↑)	AUROC (↑)	Δ_{EO} (↓)	Δ_{DP} (↓)	δ_{CF} (↓)	R^2 (↓)
Synthetic	GCN	0.686 ± 0.015	0.687 ± 0.020	0.758 ± 0.017	0.050 ± 0.030	0.060 ± 0.033	0.101 ± 0.030	0.085 ± 0.050
	GraphSAGE	0.712 ± 0.012	0.714 ± 0.021	0.789 ± 0.018	0.049 ± 0.036	0.053 ± 0.042	0.172 ± 0.056	0.011 ± 0.011
	GIN	0.682 ± 0.021	0.691 ± 0.022	0.741 ± 0.021	0.077 ± 0.053	0.081 ± 0.055	0.301 ± 0.080	0.011 ± 0.009
	C-ENC	0.665 ± 0.023	0.671 ± 0.031	0.732 ± 0.028	0.030 ± 0.024	0.048 ± 0.026	0.633 ± 0.013	0.085 ± 0.016
	FairGNN	0.668 ± 0.020	0.672 ± 0.026	0.735 ± 0.022	**0.025 ± 0.021**	**0.042 ± 0.033**	0.678 ± 0.014	0.091 ± 0.021
	NIFTY-GCN	0.618 ± 0.035	0.640 ± 0.037	0.672 ± 0.042	0.172 ± 0.110	0.199 ± 0.106	0.208 ± 0.090	0.105 ± 0.081
	NIFTY-SAGE	0.664 ± 0.041	0.682 ± 0.073	0.755 ± 0.021	0.031 ± 0.027	0.048 ± 0.027	0.147 ± 0.071	0.008 ± 0.005
	GEAR	**0.718 ± 0.018**	**0.724 ± 0.022**	**0.793 ± 0.014**	0.052 ± 0.038	0.064 ± 0.038	**0.002 ± 0.002**	**0.007 ± 0.006**
Bail	GCN	0.838 ± 0.017	0.782 ± 0.023	0.885 ± 0.018	0.023 ± 0.019	0.075 ± 0.014	0.132 ± 0.059	0.075 ± 0.028
	GraphSAGE	**0.854 ± 0.026**	**0.804 ± 0.032**	**0.905 ± 0.021**	0.039 ± 0.022	0.086 ± 0.039	0.088 ± 0.047	0.069 ± 0.011
	GIN	0.731 ± 0.058	0.656 ± 0.084	0.773 ± 0.069	0.041 ± 0.023	0.065 ± 0.034	0.143 ± 0.069	0.047 ± 0.036
	C-ENC	0.842 ± 0.047	0.792 ± 0.014	0.889 ± 0.033	0.038 ± 0.022	0.069 ± 0.020	0.040 ± 0.025	0.078 ± 0.024
	FairGNN	0.835 ± 0.024	0.784 ± 0.021	0.882 ± 0.035	0.046 ± 0.013	0.074 ± 0.026	0.042 ± 0.032	0.086 ± 0.016
	NIFTY-GCN	0.752 ± 0.065	0.669 ± 0.050	0.799 ± 0.051	0.019 ± 0.015	**0.036 ± 0.022**	0.031 ± 0.017	0.025 ± 0.018
	NIFTY-SAGE	0.823 ± 0.048	0.723 ± 0.103	0.876 ± 0.043	**0.014 ± 0.006**	0.047 ± 0.015	0.013 ± 0.011	0.044 ± 0.020
	GEAR	0.852 ± 0.026	0.800 ± 0.031	0.896 ± 0.016	0.019 ± 0.023	0.058 ± 0.017	**0.003 ± 0.002**	**0.038 ± 0.012**

[1] Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.

Overview of Part III

Other Fairness on Graphs

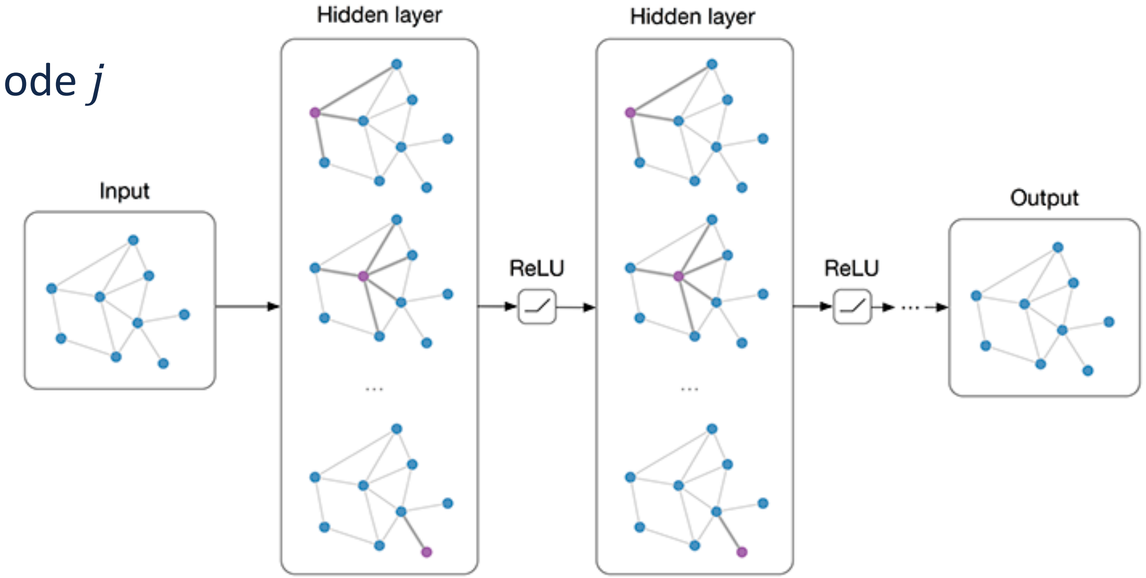
- **Counterfactual Fairness**
  - NIFTY
  - GEAR
- **Degree Fairness**
  - SL-DSGCN
  - RawlsGCN

Recap: Graph Convolutional Network (GCN)

- **Key idea:** iteratively performing neighborhood aggregation for node representation learning
- **Formulation:** graph convolution

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)}\left(\sum_{j\in\mathcal{N}_i\cup\{i\}} a_{ij}\mathbf{h}_j^{(l)}\right)\right)$$

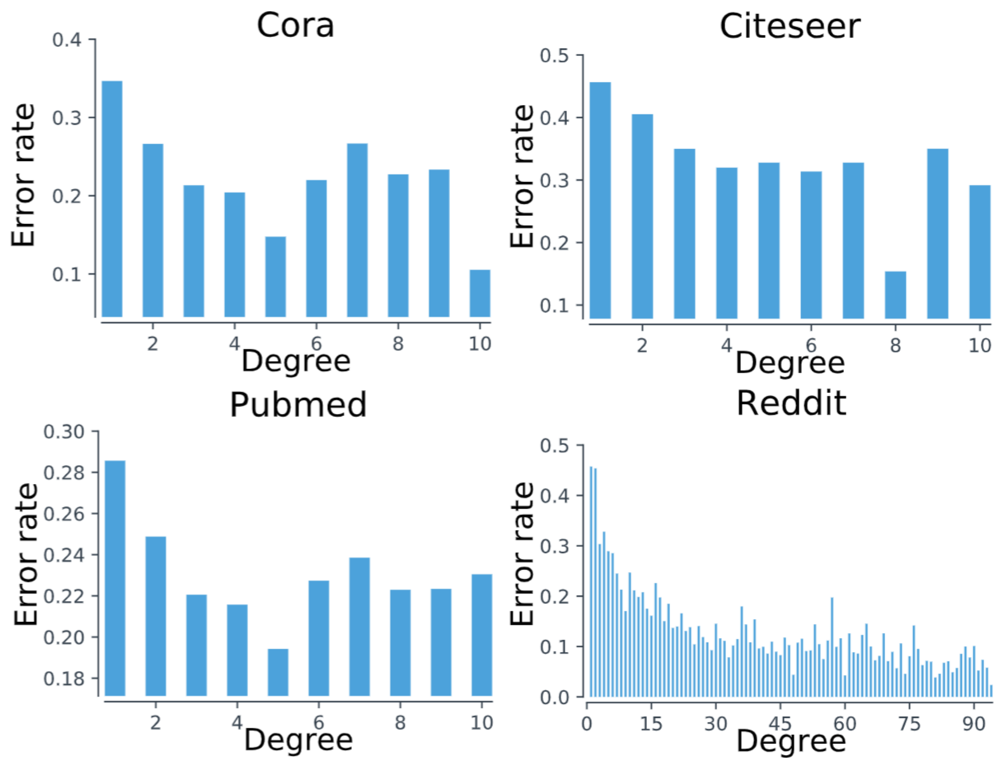
  - $\mathbf{h}_j^{(l)}$: the representation of node j at l-th layer
  - $\mathbf{W}^{(l)}$: weight parameters at l-th layer
  - $a_{ij} = \frac{1}{\sqrt{d_i+1}\sqrt{d_j+1}}$: weight of the edge between node i w.r.t. node j
  - d_i, d_j: degree of node i and node j, respectively
  - $\mathcal{N}_i$: neighborhood of node i



[1] Kipf, T. N., & Welling, M.. Semi-supervised Classification with Graph Convolutional Networks. ICLR 2017.

GCN Analysis: Error Rate vs. Node Degree

- **Observation:** low-degree nodes get higher error rate

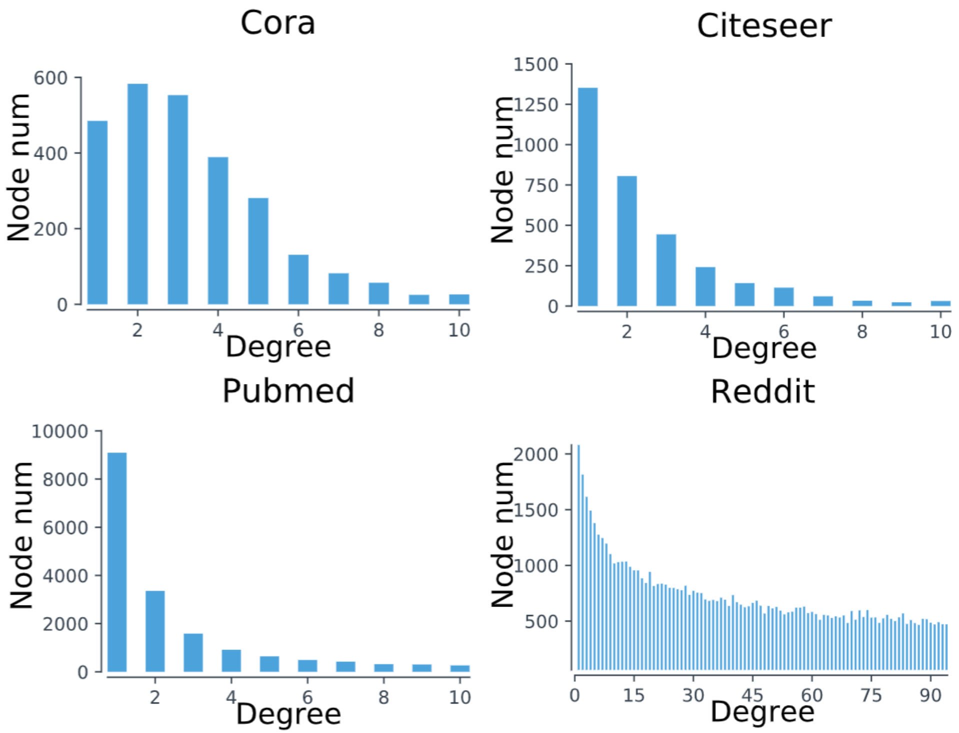


- **Questions**
  - Why is the correlation between error rate and degree bad?
  - why should we concern about low-degree nodes?

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Degree Distributions of Real-world Graphs

- Degree distribution is often long-tailed



- GCN might
  - Benefit a relatively small fraction of high-degree nodes
  - Overlook a relatively large fraction of low-degree nodes

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

GCN Limitations: Degree-related Bias

- **Key steps in GCN training**
  - Learn node representations by message passing
  - Train the model parameters by backpropagation
- **Question #1:** does GCN fail because of the message passing schema?
  - **Hypothesis #1:** high-degree nodes have higher influence to affect the training of GCN on other nodes
- **Question #2:** does GCN fail during the backpropagation?
  - Only information of labeled nodes can be backpropagated to its neighbors
  - **Hypothesis #2:** high-degree nodes are more likely to connect with labeled nodes

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Hypothesis #1: Influence of High-Degree Nodes

- **Given**
  - $\mathcal{V}_{\text{labeled}}$: a set of labeled nodes $\mathcal{V}_{\text{labeled}}$
  - $\mathbf{W}^{(L)}$: the weight of L-th layer in an L-layer GCN
  - d_i: degree of node i
  - $\mathbf{x}_i$: input node feature of node i
  - $\mathbf{h}_i^{(L)}$: output embeddings of node i learned by the L-layer GCN
- **Influence of node $\boldsymbol{i}$ to node $\boldsymbol{k}$**

$$\mathbb{E}\left[\partial \mathbf{h}_i^{(L)} / \partial \mathbf{x}_k\right] \propto \sqrt{d_i d_k} \mathbf{W}^{(L)}$$

- **Influence of node $\boldsymbol{i}$ on GCN training**

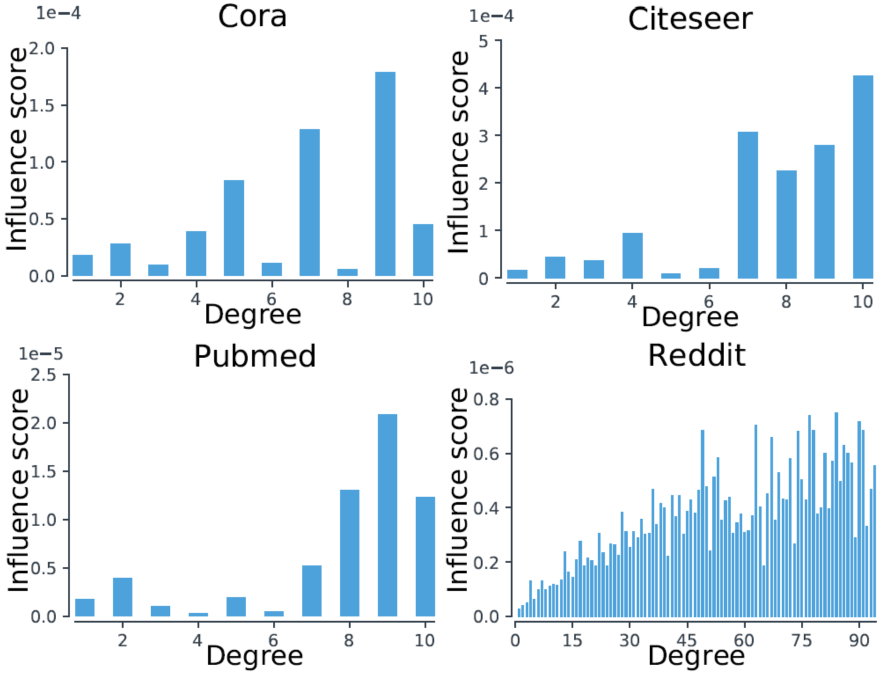
$$S(i) = \sum_{k \in \mathcal{V}_{\text{labeled}}} \left\| \mathbb{E}\left[\partial \mathbf{h}_i^{(L)} / \partial \mathbf{x}_k\right] \right\| \propto \sqrt{d_i} \left\| \mathbf{W}^{(L)} \right\| \sum_{k \in \mathcal{V}_{\text{labeled}}} \sqrt{d_k}$$

- **Remark**
  - For two nodes i and j, if $d_i > d_j$, then $S(i) > S(j)$
    → Node with higher degree will have higher influence on GCN training

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Hypothesis #1: Visualization of Node Influence

- **Goal:** visualize the influence score $S(\cdot)$ for each node
- **Observation:** high-degree nodes have higher influence score

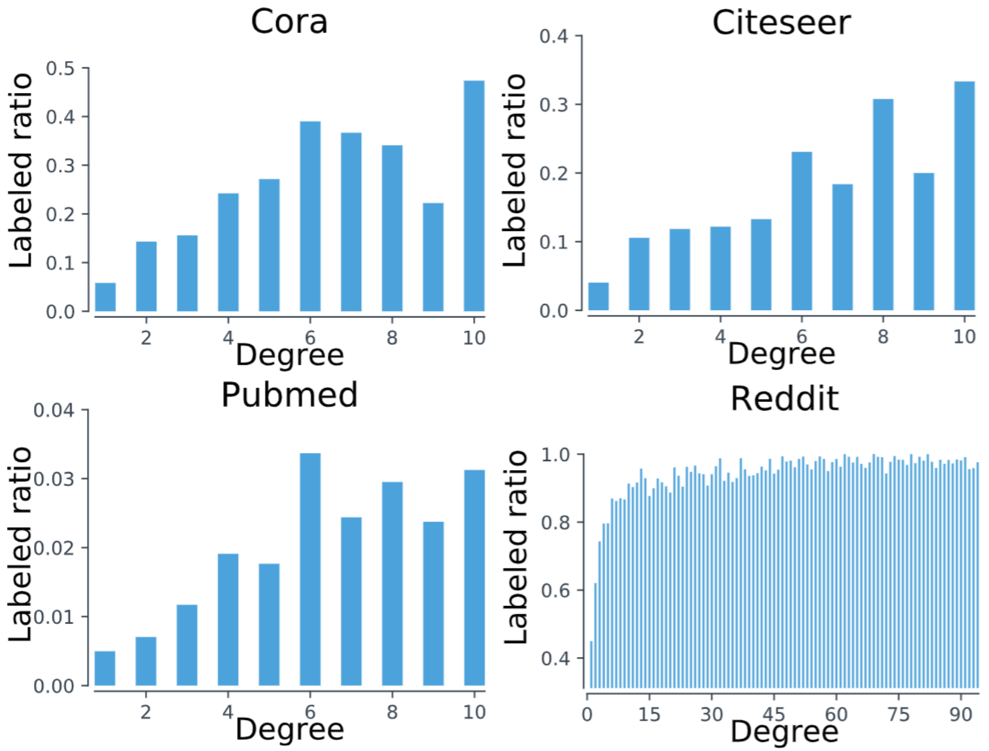


- **Question #1:** how to mitigate the impact of node degree?

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Hypothesis #2: Ratio of Labeled Neighbors

- **Observation:** high-degree nodes are more likely to have labeled neighbors

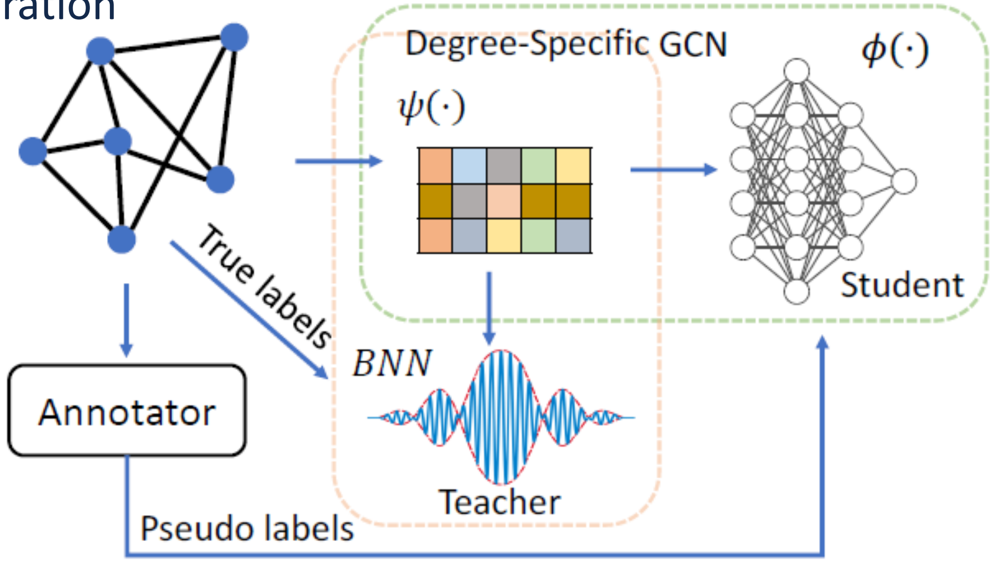


- **Question #2:** how to ensure enough training signals for low-degree nodes receive

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

SL-DSGCN: Framework

- **Strategy:** pre-training + fine-tuning
- **Pre-training**
  - Mitigate the impact of node degree by degree-specific GCN
  - Pre-train
    - A Bayesian neural network (BNN) with true labels for further use during fine-tuning
    - An annotator through label propagation for pseudo-label generation



[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Degree-specific Graph Convolutional Network (DSGCN)

- **Key components**
  - A stack of degree-specific graph convolution layer for embedding learning
  - A fully-connected layer for node classification
- **Given:** the settings of l-th graph convolution layer and
  - d_i: the degree of node i
  - $\mathbf{W}_{d_j}^{(l)}$: the degree-specific weight w.r.t. degree of node j
- **Degree-specific graph convolution layer**

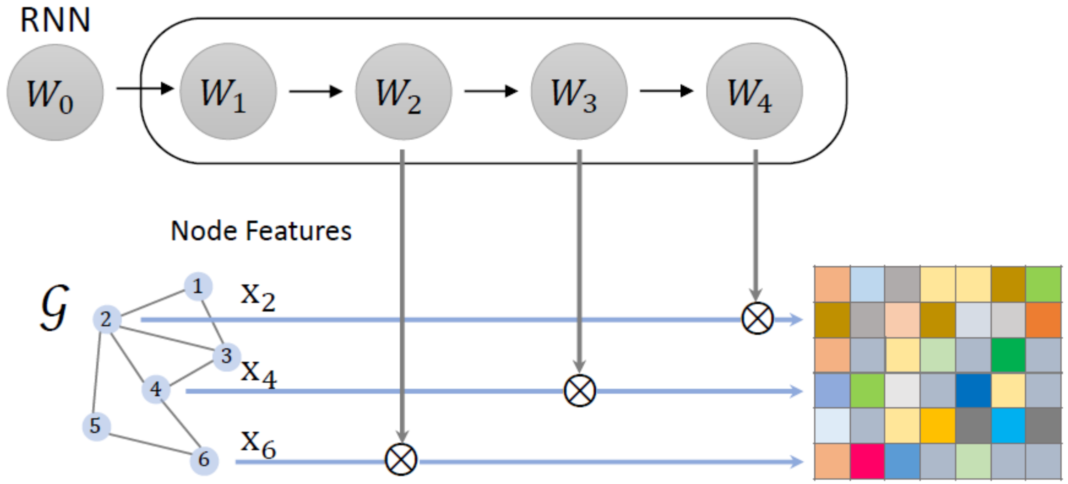
$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} a_{ij} \left(\mathbf{W}^{(l)} + \mathbf{W}_{d_j}^{(l)}\right) \mathbf{h}_j^{(l)}\right)$$

- **Question:** how to generate the degree-specific weight?

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Degree-specific Weight Generation

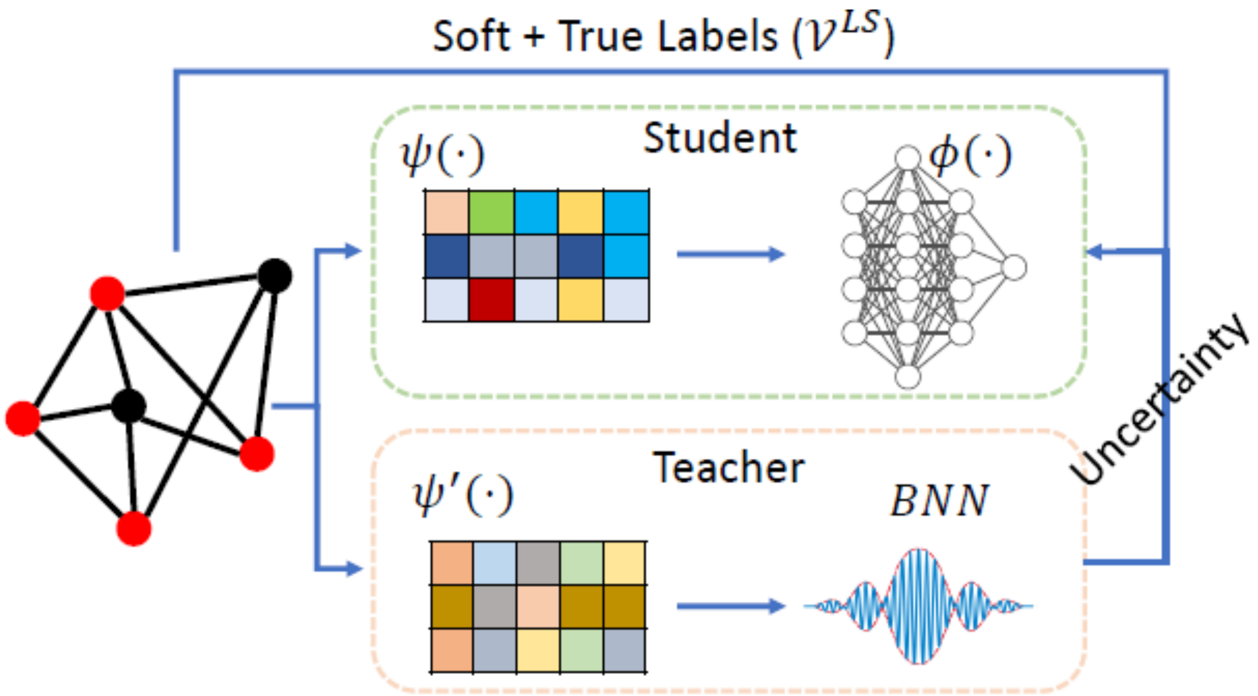
- **Hypothesis:** existence of the complex relations among nodes with different degrees
- **Method:** weight generation with recurrent neural network (RNN)
- **Given**
  - A RNN
  - $\mathbf{W}_k^{(l)}$ = degree-specific weight of degree k at l-th layer
- Weight of degree $k+1$ at l-th layer is $\mathbf{W}_{k+1}^{(l)} = \mathrm{RNN}\left(\mathbf{W}_k^{(l)}\right)$



[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

SL-DSGCN: Framework

- **Strategy:** pre-training + fine-tuning
- **Fine-tuning**
  – Provide pseudo training signals to low-degree nodes for self-supervision

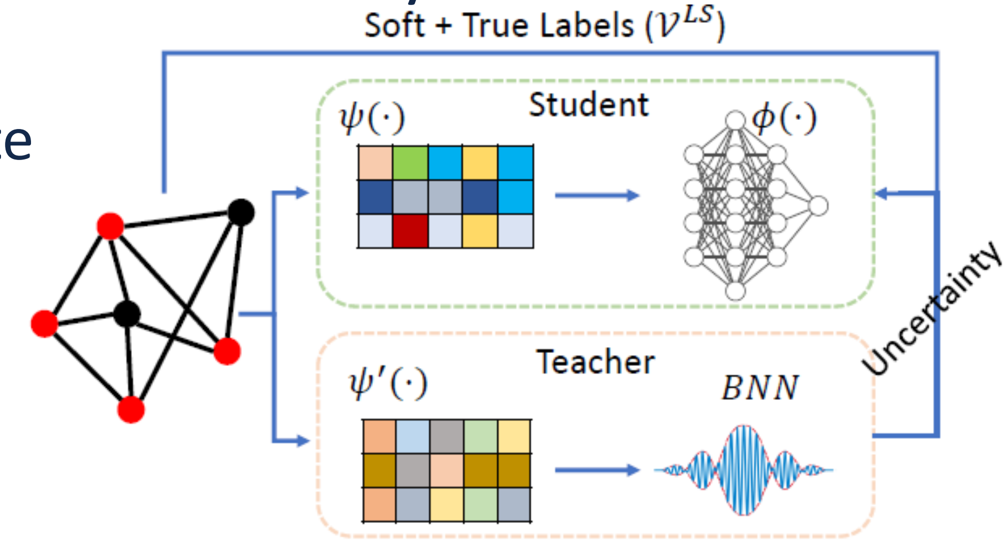


[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Fine-Tuning with Self-Supervised Learning

- **Student network:** degree-specific GCN (DSGCN)
- **Teacher network:** Bayesian neural network (BNN)
  - Provide additional softly-labeled set for self-supervision in student network

    Nodes labeled identically by the pseudo-label annotator and BNN
  - Exponentially decay the learning rate of labeled and softly-labeled nodes by uncertainty score
    - Higher uncertainty score → smaller learning rate



[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

SL-DSGCN: Effectiveness Results

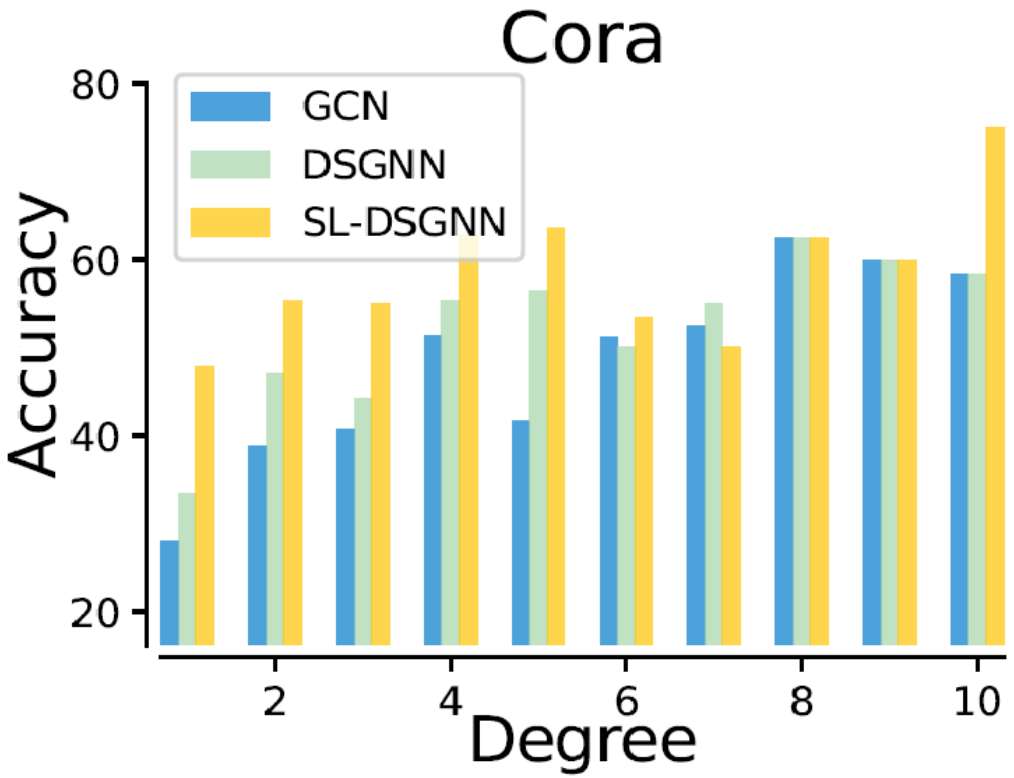
- **Observations**
  - Increased label rate implies higher classification accuracy
  - Self-supervision provides useful information (i.e., high accuracy when the label rate is low)
  - SL-DSGCN outperforms all baseline methods

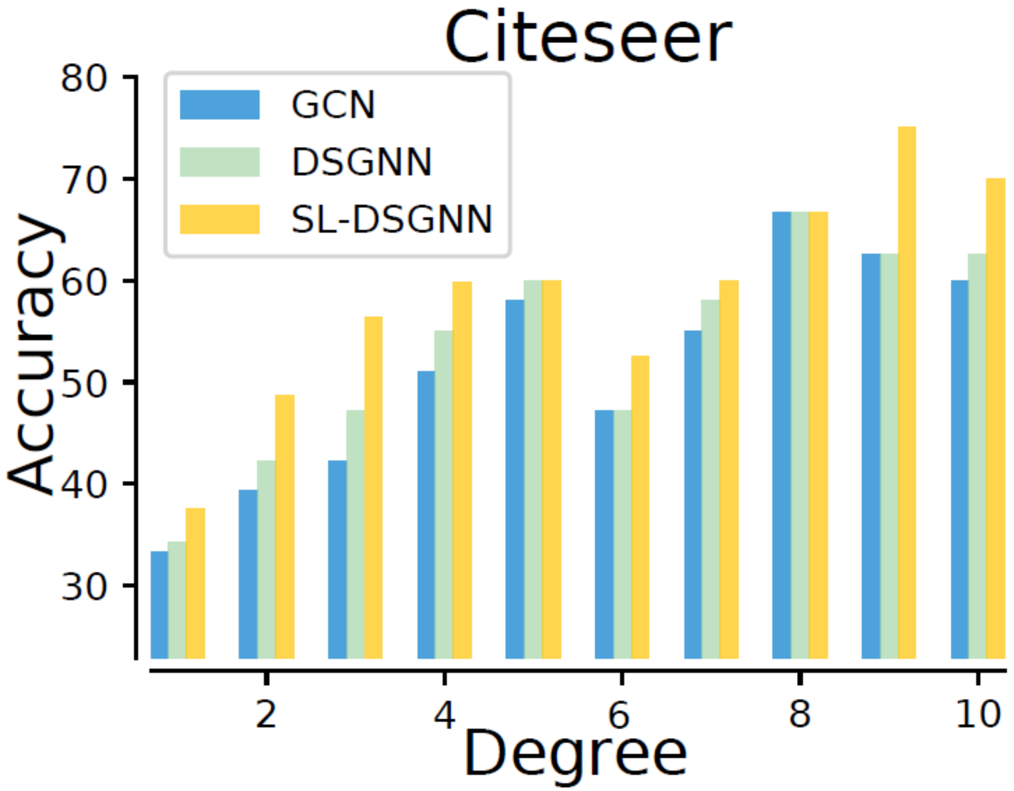
Dataset	Cora					Citeseer					PubMed		
Label Rate	0.5%	1%	2%	3%	4%	0.5%	1%	2%	3%	4%	0.03%	0.06%	0.09%
LP	29.05	38.63	53.26	70.31	73.47	32.10	40.08	42.83	45.32	49.01	39.01	48.7	56.73
ParWalks	37.01	41.40	50.84	58.24	63.78	19.66	23.70	29.17	35.61	42.65	35.15	40.27	51.33
GCN	35.89	46.00	60.00	71.15	75.68	34.50	43.94	54.42	56.22	58.71	47.97	56.68	63.26
DEMO-Net	33.56	40.05	61.18	72.80	77.11	36.18	43.35	53.38	56.5	59.85	48.15	57.24	62.95
Self-Train	43.83	52.45	63.36	70.62	77.37	42.60	46.79	52.92	58.37	60.42	57.67	61.84	64.73
Co-Train	40.99	52.08	64.27	73.04	75.86	40.98	56.51	52.40	57.86	62.83	53.15	59.63	65.50
Union	45.86	53.59	64.86	73.28	77.41	45.82	54.38	55.98	60.41	59.84	58.77	60.61	67.57
Interesction	33.38	49.26	62.58	70.64	77.74	36.23	55.80	56.11	58.74	62.96	59.70	60.21	63.97
M3S	50.28	58.74	68.04	75.09	78.80	48.96	53.25	58.34	61.95	63.03	59.31	65.25	70.75
SL-DSGCN	**53.58**	**61.36**	**70.31**	**80.15**	**81.05**	**54.07**	**56.68**	**59.93**	**62.20**	**64.45**	**61.15**	**65.68**	**71.78**

[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

SL-DSGCN: Fairness Results

- **Observations:** degree-wise classification accuracy
  - SL-DSGCN > DSGNN > GCN for all degrees, especially low degrees





[1] Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.

Overview of Part III

Other Fairness on Graphs

Counterfactual Fairness
- NIFTY
- GEAR

Degree Fairness
- SL-DSGCN
- RawlsGCN

Limitations: SL-DSGCN

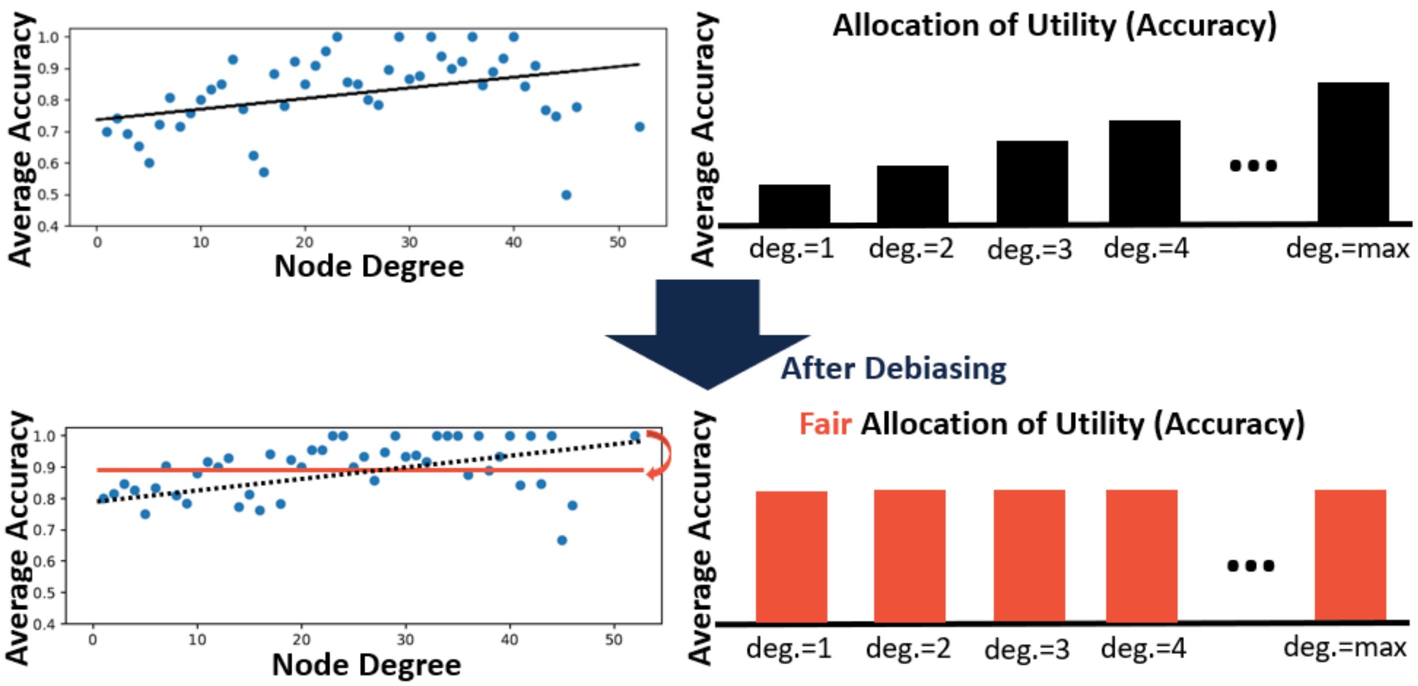
- **SL-DSGCN**
  - **Degree-specific weight:** learn degree-specific weights, generated by RNN
  - **Self-supervised learning:** generate pseudo labels for additional training signals
- **Limitation 1:** additional number of weight parameters
  - Weight parameters of RNN for degree-specific weight generation
- **Limitation 2:** change(s) to the GCN architecture
  - Degree-specific weight generator
  - Self-supervised learning module
- **Question:** how to mitigate degree-related unfairness without
  - Hurting the scalability of GCN
  - Changing the GCN architecture?

High cost of computational resources

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

Fairness = Just Allocation of Utility

- **Intuition:** utility = resource to allocate
- **Expected result:** similar utility (accuracy) for all nodes regardless of their degrees

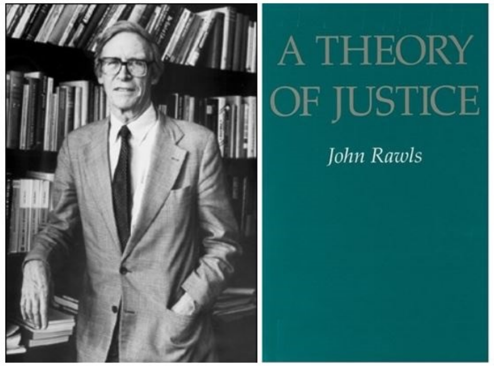


- **Question:** how to define such fairness?

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

Recap: Rawlsian Difference Principle

- **Origin:** distributive justice
- **Goal:** fairness as just allocation of social welfare



"Inequalities are permissible when they maximize […] the long-term expectations of the least fortunate group."

-- John Rawls, 1971

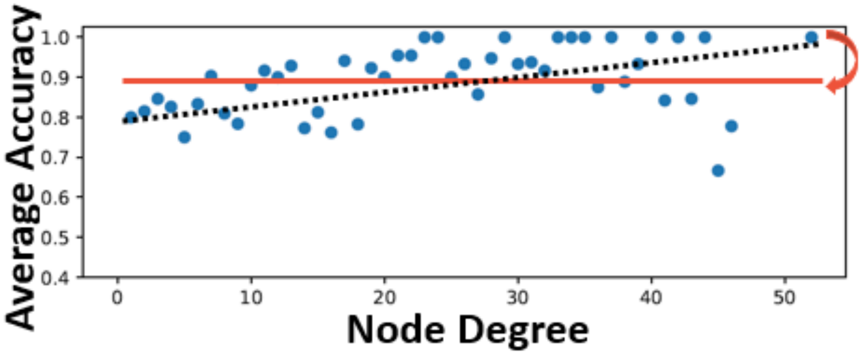
- **Intuition:** treat utility of GCN as welfare to allocate
  - Least fortunate group → group with the smallest utility
  - **Example:** classification accuracy for node classification

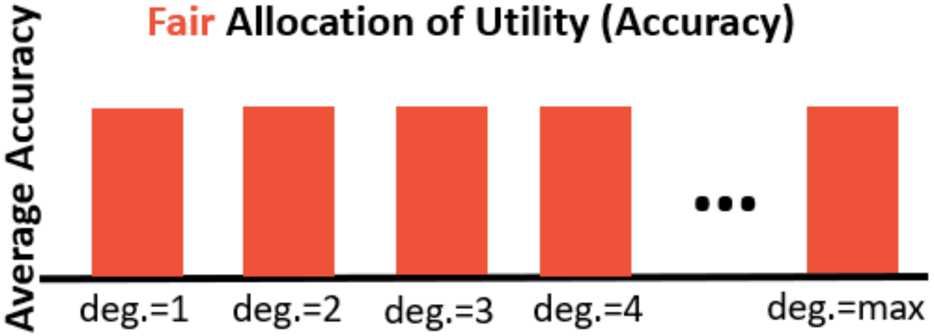
[1] Rawls, J.. A Theory of Justice. Press, Cambridge 1971.

- **Justice as fairness**
  - Justice is a virtue of instituitions
  - Free persons enjoy and acknowledge the rules
- **Well-ordered society**
  - Designed to advance the good of its members
  - Regulated by a public conception of justice

RawlsGCN: Problem Definition

- **Given**
  - $\mathcal{G} = (\mathbf{A}, \mathbf{X})$: an undirected graph
  - $\boldsymbol{\theta}$: weights of an L-layer GCN
  - J: a task-specific loss
- **Find:** a well-trained GCN that
  - Minimizes the task-specific loss
  - Achieves a fair allocation of utility for the groups of nodes with the same degree
- **Key question:** when is the allocation of utility fair?





[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

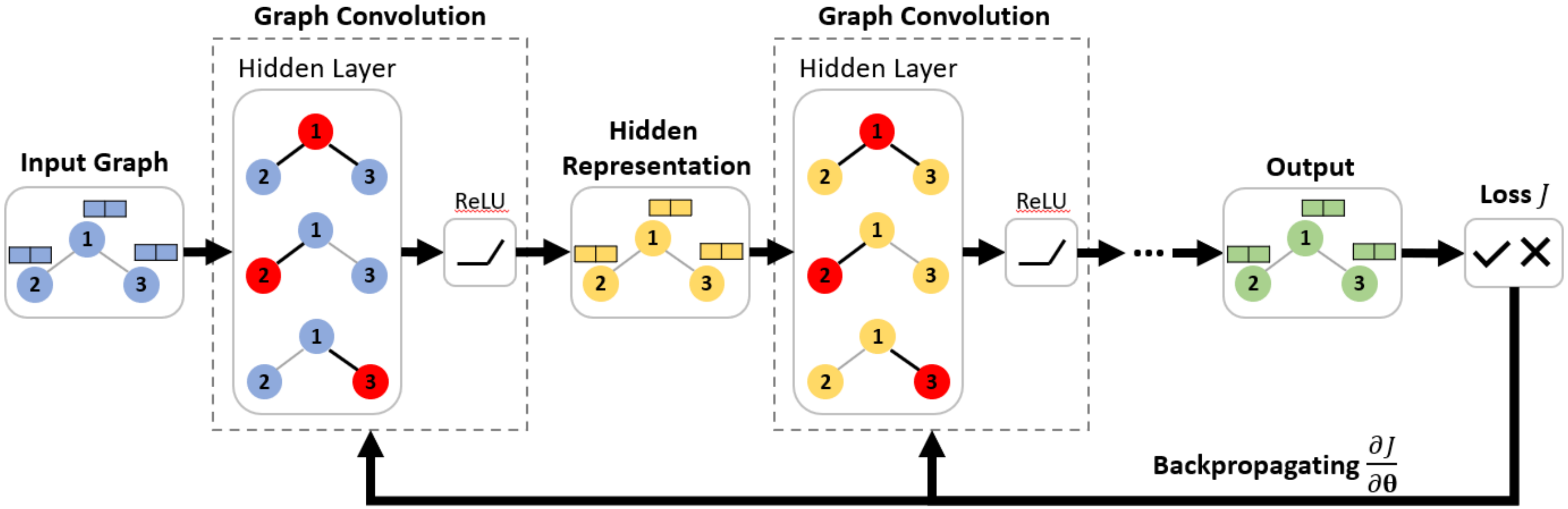
RawlsGCN: Fair Allocation of Utility

- **Key idea:** consider the stability of the Rawlsian difference principle
- **How to achieve the stability?**
  - Keep improving the utility of the least fortunate group
- **When do we achieve the stability?**
  - No least fortunate group
  - All groups have the balanced utility
- **Challenge:** non-differentiable utility
  - **Workaround:** use loss function as the proxy of utility
  - **Rationale:** minimize loss in order to maximize utility
- **Goal:** fair allocation of utility → balanced loss
- **Question:** why does the loss vary after training the GCN?

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN: Source of Unfairness

- **Intuition:** understand why the loss varies **after training**
- **What happens during training?**
  - Extract node representations and make predictions
  - Calculate the task-specific loss J
  - Update model weights $\boldsymbol{\theta}$ by the gradient $\frac{\partial J}{\partial \boldsymbol{\theta}}$ ← key component for training
- **Question:** is the unfairness caused by the gradient?



[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN: Gradient of Model Weights

- **Given**
  - An undirected graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ with $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$
  - An arbitrary l-th graph convolution layer
    - Weight matrix $\mathbf{W}^{(l)}$
    - Hidden representations before activation $\mathbf{E}^{(l)} = \hat{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}$
  - A task-specific loss J
- **The gradient of J w.r.t. $\mathbf{W}^{(l)}$**

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}} = \left(\mathbf{H}^{(l-1)}\right)^T \hat{\mathbf{A}}^T \frac{\partial J}{\partial \mathbf{E}^{(l)}}$$

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN: Unfairness in Gradient

- **Gradient of loss w.r.t. weight**

Column sum of i-th column

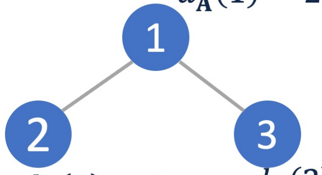
Row sum of j-th row

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}} = \sum_{i=1}^{n} d_{\hat{\mathbf{A}}}(i)\mathbb{I}_i^{(\mathrm{col})} = \sum_{j=1}^{n} d_{\hat{\mathbf{A}}}(j)\mathbb{I}_j^{(\mathrm{row})}$$

  - $\mathbb{I}_i^{(\mathrm{col})} = \left(\mathbb{E}_{j\sim\mathcal{N}(i)}\left[\mathbf{H}^{(l-1)}[j,:]\right]\right)^T \frac{\partial J}{\partial \mathbf{E}^{(l)}[i,:]}$
  - $\mathbb{I}_j^{(\mathrm{row})} = \left(\mathbf{H}^{(l-1)}[j,:]\right)^T \mathbb{E}_{\boxed{i\sim\mathcal{N}(j)}}\left[\frac{\partial J}{\partial \mathbf{E}^{(l)}[i,:]}\right]$

Sampling from j-th neighborhood

$d_{\hat{\mathbf{A}}}(1) = 2$



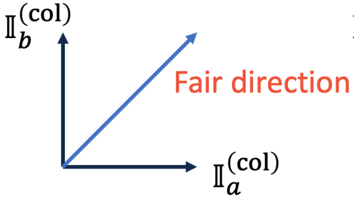
$d_{\hat{\mathbf{A}}}(2) = 1$ $d_{\hat{\mathbf{A}}}(3) = 1$

$$\frac{\partial J}{\partial \mathbf{W}^{(l)}} = \boxed{2}\mathbb{I}_1^{(\mathrm{col})} + \mathbb{I}_2^{(\mathrm{col})} + \mathbb{I}_3^{(\mathrm{col})}$$

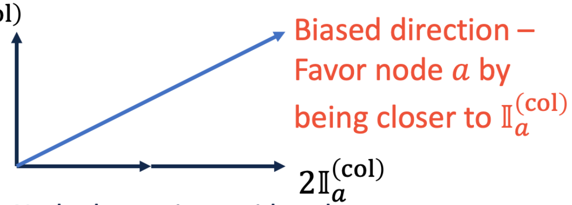
Higher importance due to higher degree

- **Intuitions**
  - $\mathbb{I}_i^{(\mathrm{col})}$ and $\mathbb{I}_j^{(\mathrm{row})}$ → The directions for gradient descent
  - $d_{\hat{\mathbf{A}}}(i)$ and $d_{\hat{\mathbf{A}}}(j)$ → The importance of the direction

Node a: $d_{\hat{\mathbf{A}}}(a) = 2$
Node b: $d_{\hat{\mathbf{A}}}(b) = 1$

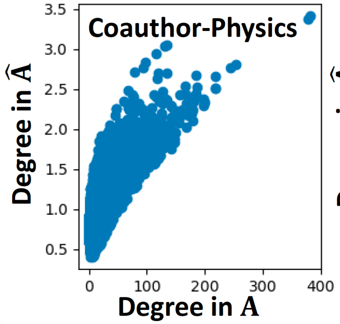


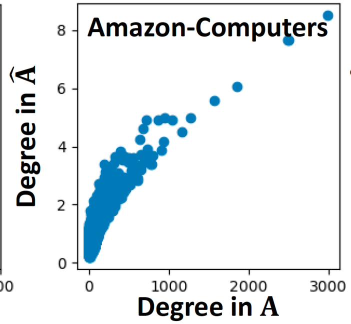
Node degree takes no effect

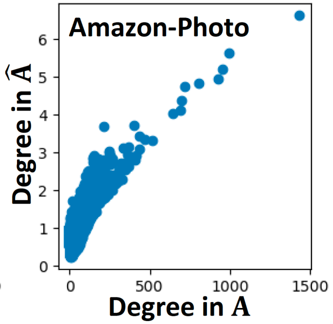


Node degree is considered

- Higher degree → more focus on that direction
- **Symmetric normalization in $\hat{\mathbf{A}}$**
  - Normalize the largest eigenvalue, not degree
  - High degree in A → high degree in $\hat{\mathbf{A}}$
- **Solution:** doubly stochastic matrix $\hat{\mathbf{A}}_{\mathrm{DS}}$







[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN: Doubly Stochastic Matrix Computation

- **How to mitigate unfairness in $\frac{\partial J}{\partial \mathbf{W}^{(l)}}$?**
  - **Intuition:** enforce row sum and column sum of $\hat{\mathbf{A}}$ to be 1
  - **Solution:** doubly stochastic normalization on $\hat{\mathbf{A}}$
- **Method:** Sinkhorn-Knopp algorithm
  - **Key idea:** iteratively normalize the row and column of a matrix
  - **Complexity:** linear time and space complexity
  - **Convergence:** always converge iff. the matrix has total support
- **Existence for GCN:** the Sinkhorn-Knopp algorithm always finds the unique doubly stochastic form $\hat{\mathbf{A}}_{\mathrm{DS}}$ of $\hat{\mathbf{A}}$
  - $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$
  - $\tilde{\mathbf{D}} =$ degree matrix of $\mathbf{A} + \mathbf{I}$ for a graph $\mathbf{A}$

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN: A Family of Debiasing Methods

- **Gradient computation**

$$\left(\frac{\partial J}{\partial \mathbf{W}^{(l)}}\right)_{\text{fair}} = \left(\mathbf{H}^{(l-1)}\right)^T \hat{\mathbf{A}}_{\text{DS}}^T \frac{\partial J}{\partial \mathbf{E}^{(l)}}$$

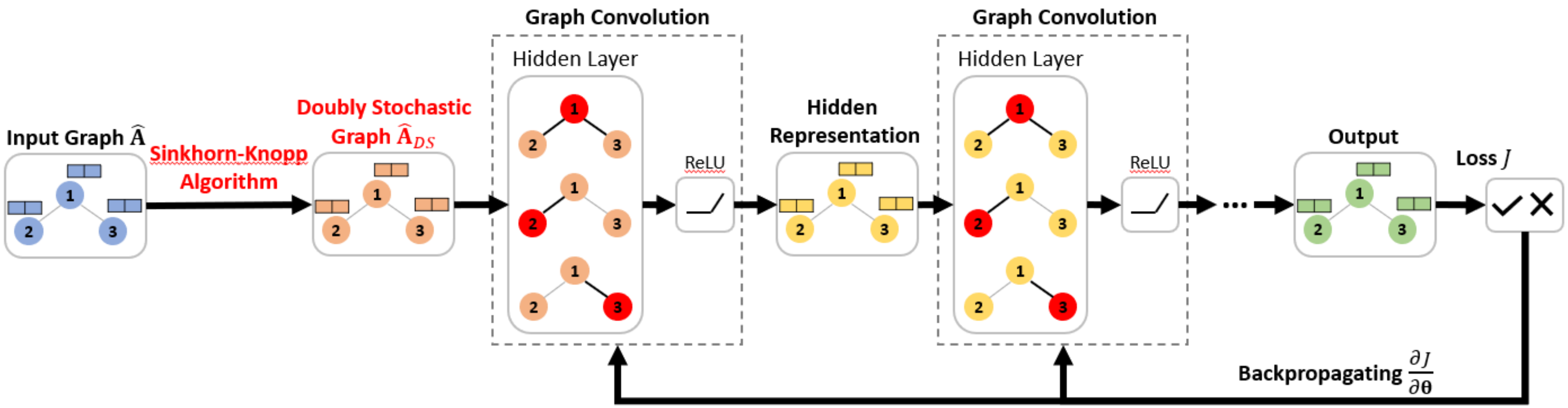
  - **Key term: $\hat{\mathbf{A}}_{\text{DS}}$** – doubly-stochastic normalization of $\hat{\mathbf{A}}$

- **Proposed methods**
  - **RawlsGCN-Graph:** during data pre-processing, compute $\hat{\mathbf{A}}_{\text{DS}}$ and treat it as the input of GCN
  - **RawlsGCN-Grad:** during optimization (in-processing), treat $\hat{\mathbf{A}}_{\text{DS}}$ as a normalizer to equalize the importance of node influence

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN-Graph: Pre-processing

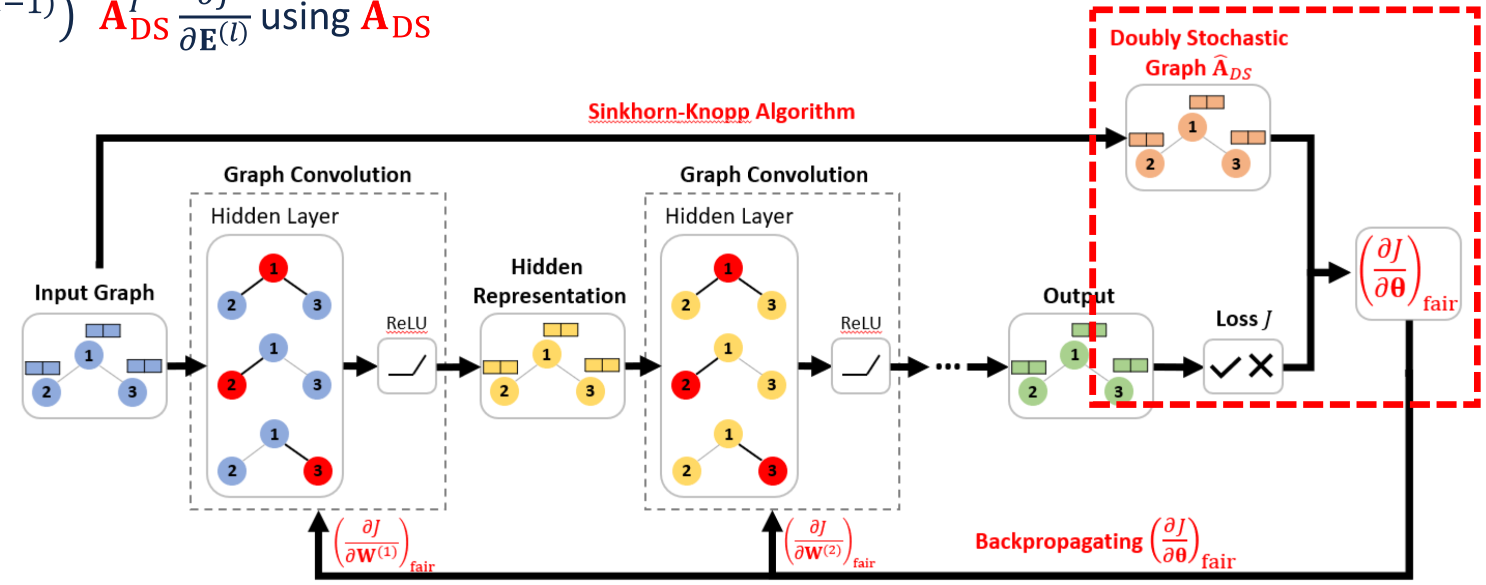
- **Intuition:** normalize the input renormalized graph Laplacian into a doubly stochastic matrix
- **Key steps**
  1. Precompute the renormalized graph Laplacian $\hat{\mathbf{A}}$
  2. Precompute $\hat{\mathbf{A}}_{\mathrm{DS}}$ by applying the Sinkhorn-Knopp algorithm
  3. Input $\hat{\mathbf{A}}_{\mathrm{DS}}$ and $\mathbf{X}$ (node features) to GCN for training



[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

RawlsGCN-Grad: In-processing

- **Intuition:** equalize the importance of node influence in gradient computation
- **Key steps**
  1. Precompute the renormalized graph Laplacian $\hat{\mathbf{A}}$
  2. Input $\hat{\mathbf{A}}$ and $\mathbf{X}$ (node features) to GCN
  3. Compute $\hat{\mathbf{A}}_{\mathrm{DS}}$ by applying the Sinkhorn-Knopp algorithm
  4. Repeat until maximum number of training epochs
     - Compute the fair gradient $\left(\frac{\partial J}{\partial \mathbf{W}^{(l)}}\right)_{\text{fair}} = \left(\mathbf{H}^{(l-1)}\right)^T \hat{\mathbf{A}}_{\mathrm{DS}}^T \frac{\partial J}{\partial \mathbf{E}^{(l)}}$ using $\hat{\mathbf{A}}_{\mathrm{DS}}$
     - Update $\mathbf{W}^{(l)}$ by the fair gradient $\left(\frac{\partial J}{\partial \mathbf{W}^{(l)}}\right)_{\text{fair}}$



RawlsGCN: Effectiveness Results

- **Observations**
  - RawlsGCN achieves the smallest bias
  - Classification accuracy can be improved
    - Mitigating the bias → higher accuracy for low-degree nodes → higher overall accuracy

Method	Coauthor-Physics		Amazon-Computers		Amazon-Photo	
	Acc.	Bias	Acc.	Bias	Acc.	Bias
GCN	93.96 ± 0.367	0.023 ± 0.001	64.84 ± 0.641	0.353 ± 0.026	79.58 ± 1.507	0.646 ± 0.038
DEMO-Net	77.50 ± 0.566	0.084 ± 0.010	26.48 ± 3.455	0.456 ± 0.021	39.92 ± 1.242	0.243 ± 0.013
DSGCN	79.08 ± 1.533	0.262 ± 0.075	27.68 ± 1.663	1.407 ± 0.685	26.76 ± 3.387	0.921 ± 0.805
Tail-GNN	OOM	OOM	76.24 ± 1.491	1.547 ± 0.670	86.00 ± 2.715	0.471 ± 0.264
AdvFair	87.44 ± 1.132	0.892 ± 0.502	53.50 ± 5.362	4.395 ± 1.102	75.80 ± 3.563	51.24 ± 39.94
REDRESS	94.48 ± 0.172	0.019 ± 0.001	80.36 ± 0.206	0.455 ± 0.032	89.00 ± 0.369	0.186 ± 0.030
RawlsGCN-Graph (Ours)	94.06 ± 0.196	0.016 ± 0.000	80.16 ± 0.859	0.121 ± 0.010	88.58 ± 1.116	0.071 ± 0.006
RawlsGCN-Grad (Ours)	94.18 ± 0.306	0.021 ± 0.002	74.18 ± 2.530	0.195 ± 0.029	83.70 ± 0.672	0.186 ± 0.068

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

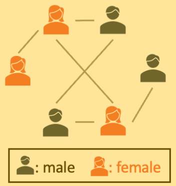
RawlsGCN: Efficiency Results

- **Observation:** RawlsGCN has the best efficiency compared with other baseline methods
  – Same number of parameters and memory usage (in MB) with GCN
  – Much shorter training time (in seconds)

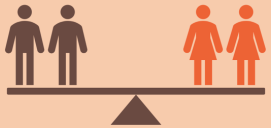
Method	# Param.	Memory	Training Time
GCN (100 epochs)	48,264	1,461	13.335
GCN (200 epochs)	48,264	1,461	28.727
DEMO-Net	11,999,880	1,661	9158.5
DSGCN	181,096	2,431	2714.8
Tail-GNN	2,845,567	2,081	94.058
AdvFair	89,280	1,519	148.11
REDRESS	48,264	1,481	291.69
RawlsGCN-Graph (Ours)	48,264	1,461	11.783
RawlsGCN-Grad (Ours)	48,264	1,461	12.924

[1] Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

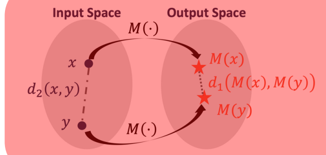
Roadmap



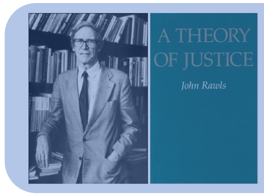
Introduction



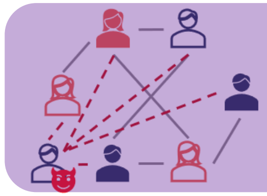
Part I: Group Fairness on Graphs



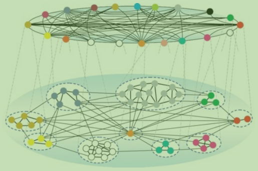
Part II: Individual Fairness on Graphs



Part III: Other Fairness on Graphs



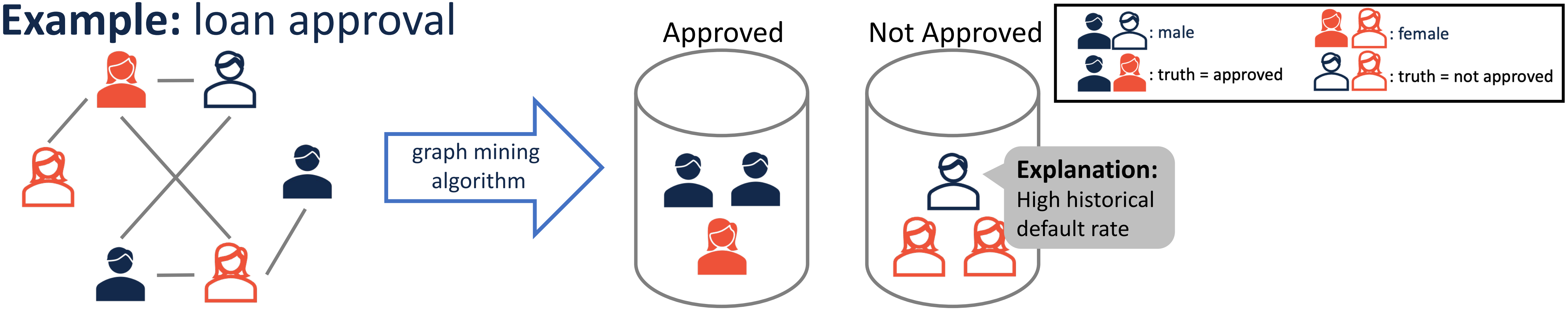
Part IV: Beyond Fairness on Graphs



Part V: Future Trends

Related Problem #1: Explainability

- **Motivation:** **how** to provide human understandable explanation to a particular prediction?
- **Goal:** explain model prediction to non-expert end users
- **Example:** loan approval

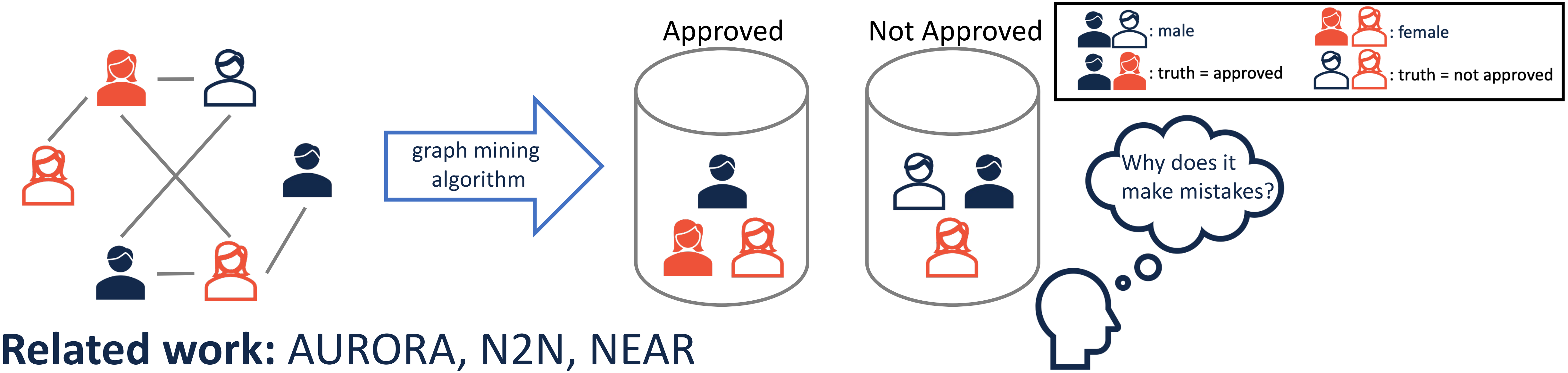


- **Related work:** GNNExplainer, PGM-Explainer, SubgraphX
- **Relationship to fairness:** explainability helps interpret whether a model uses biased information for prediction to end users

[1] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J.. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.
[2] Vu, M. N., & Thai, M. T.. PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. NeurIPS 2020.
[3] Yuan, H., Yu, H., Wang, J., Li, K., & Ji, S.. On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

Related Problem #2: Accountability

- **Motivation:** **how** do mining results relate to graph topology?
- **Goal:** find influential elements w.r.t. the graph mining results
- **Example:** loan approval

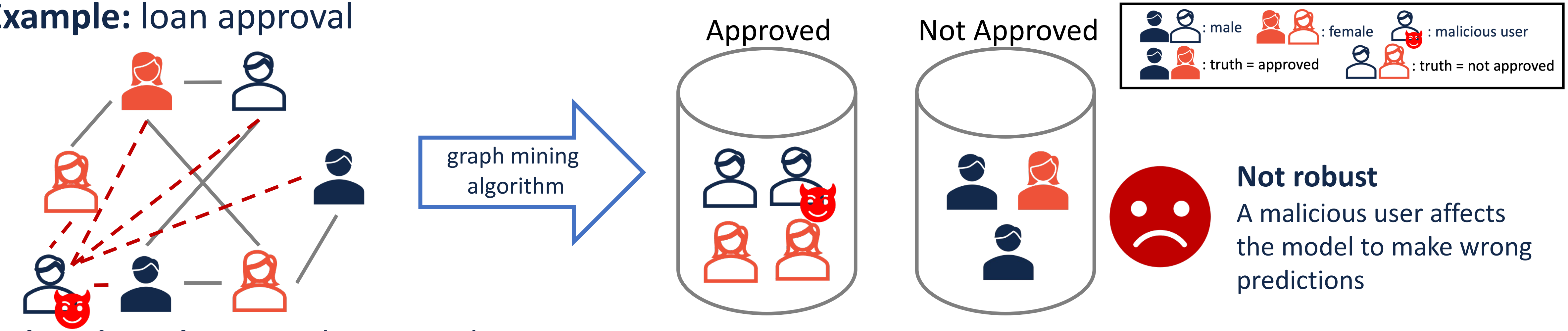


- **Related work:** AURORA, N2N, NEAR
- **Relationship to fairness:** accountability helps determine to what extent a sensitive attribute influences the graph mining results

[1] Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H.. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.
[2] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.
[3] Wang, Y., Yao, Y., Tong, H., Xu, F., & Lu, J.. Auditing Network Embedding: An Edge Influence based Approach. TKDE 2021.

Related Problem #3: Robustness

- **Motivation: why** do mining results sensitive to malicious manipulations?
- **Goals**
  - **Attack:** fool the mining model by a few manipulations on the input graph
  - **Defense:** defend the mining model against the malicious manipulations
- **Example:** loan approval



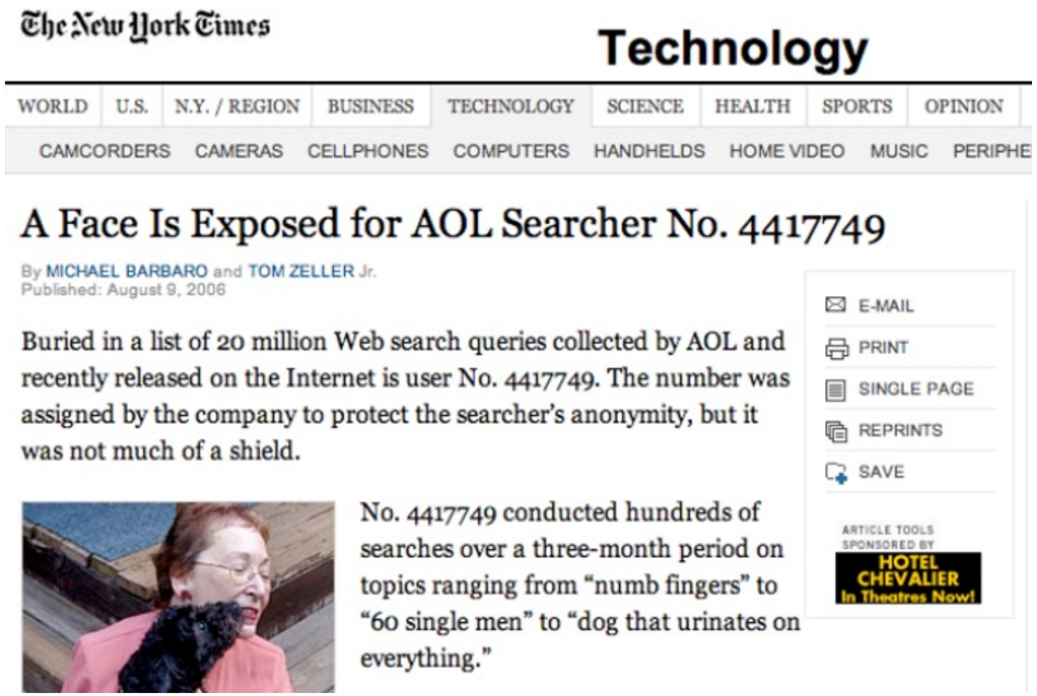
Not robust
A malicious user affects the model to make wrong predictions

- **Related work:** Nettack, Mettack, GNN-SVD
- **Relationship to fairness:** malicious users can
  - Manipulate the private sensitive information of other users
  - Attack the model to make a fair mining model biased

[1] Zügner, D., Akbarnejad, A., & Günnemann, S.. Adversarial Attacks on Neural Networks for Graph Data. KDD 2018.
[2] Zügner, D., & Günnemann, S.. Adversarial Attacks on Graph Neural Networks via Meta Learning. ICLR 2019.
[3] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., & Papalexakis, E. E.. All You Need is Low (Rank): Defending Against Adversarial Attacks on Graphs. WSDM 2020.

Related Problem #4: Privacy Preservation

- **Motivation:** **why** can we infer private information by data analysis?
- **Goal:** prevent the data or mining model from leaking private information
- **Example**



The New York Times

Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

CAMCORDERS | CAMERAS | CELLPHONES | COMPUTERS | HANDHELDS | HOME VIDEO | MUSIC | PERIPHE

A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.

No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."

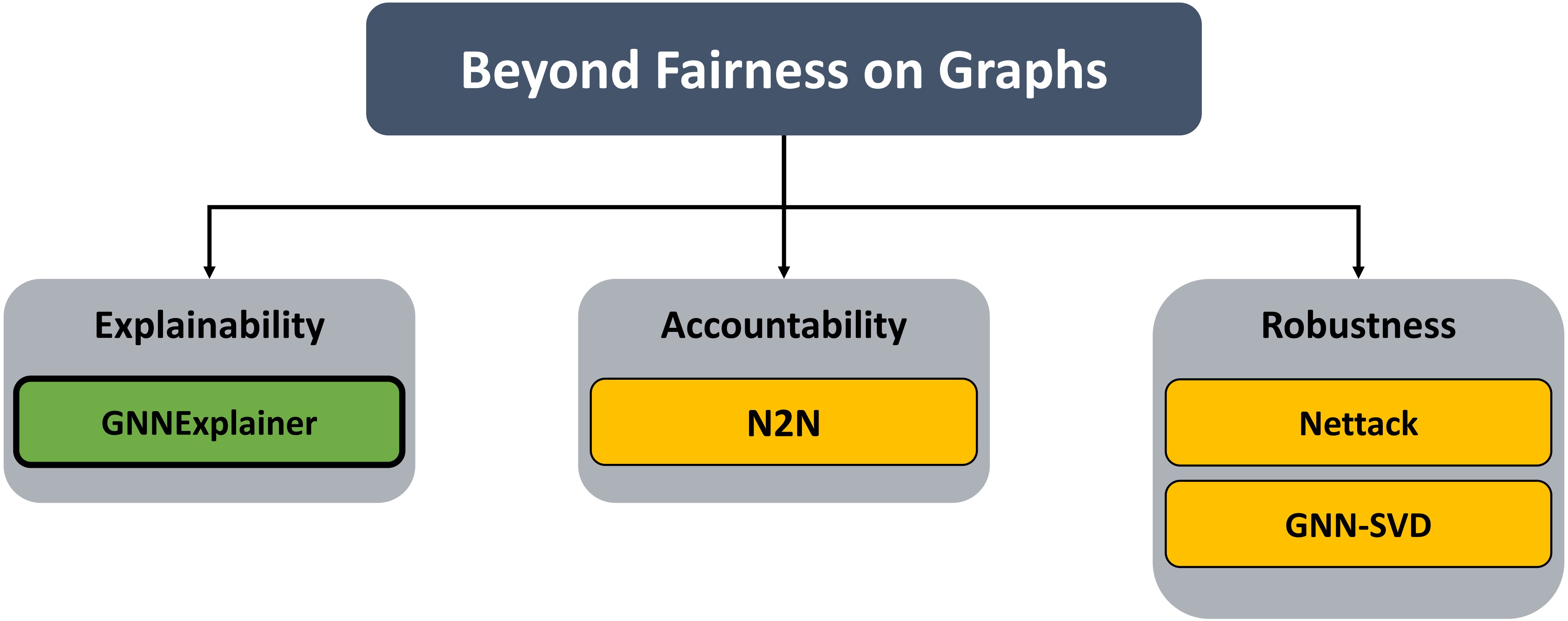
E-MAIL
PRINT
SINGLE PAGE
REPRINTS
SAVE

- AOL releases anonymized search logs of 650k users
- People find out the identity of one searcher using her search logs in a few days

- **Related work:** T_λ, dK-graph, VFGNN
- **Relationship to fairness:** preserving privacy on sensitive information may help ensure fairness

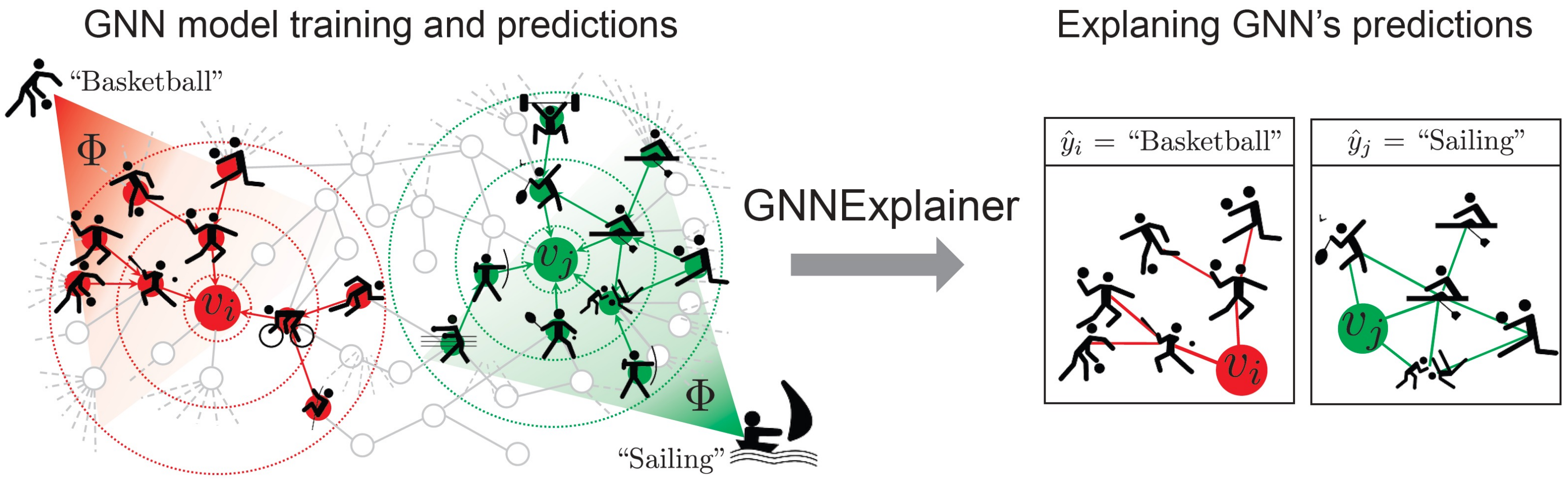
[1] Ding, X., Zhang, X., Bao, Z., & Jin, H.. Privacy-Preserving Triangle Counting in Large Graphs. CIKM 2018.
[2] Wang, Y., & Wu, X.. Preserving Differential Privacy in Degree-Correlation based Graph Generation. TDP 2013.
[3] Zhou, J., Chen, C., Zheng, L., Wu, H., Wu, J., Zheng, X., ... & Wang, L.. Vertically Federated Graph Neural Network for Privacy-Preserving Node Classification. *arXiv 2020*.

Overview of Part IV



Related Problem #1: Explainability

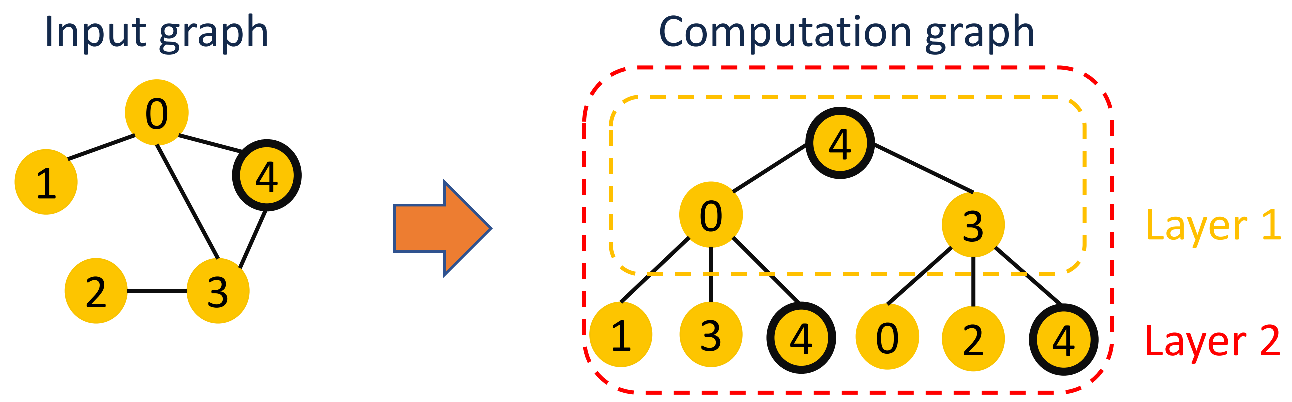
- **Observation:** graph neural network (GNN) is not transparent to end users
  - Complex neighborhood aggregation + feature transformation
  - Nonlinear activation
- **Question:** can we explain why GNN makes a certain prediction to node?
- **Representative solution:** GNNExplainer



[1] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J.. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.

GNNExplainer: Overview

- **Intuition:** find the most informative subgraph and subset of node features w.r.t. a node's prediction
  - **Reason:** GNN use feature and local subgraph to learn node representations
- **Computation graph:** a subgraph with all information about making a prediction
- **Example:** 2-layer GCN



[1] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J.. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.

GNNExplainer: Solution

- **Optimization problem**

$$\max_{G_S, \mathbf{X}_S} MI\big(\mathbf{Y}[i,:], (G_S, \mathbf{X}_S)\big) = H(\mathbf{Y}[i,:]) - H(\mathbf{Y}[i,:] | G = G_S, \mathbf{X} = \mathbf{X}_S)$$

  - $\mathbf{Y}[i,:]$: model prediction for node i
  - G_S: node i's sub-computation graph
  - $\mathbf{X}_S$: node i's subset of node features
  - $H(\mathbf{Y}[i,:])$: constant, entropy of model prediction
  - $H(\mathbf{Y}[i,:] | G = G_S, \mathbf{X} = \mathbf{X}_S)$: conditional entropy given the input subgraph and features
- **Surrogate problem**

$$\min_{\mathbf{M},\mathbf{F}} H(\mathbf{Y}[i,:] | \mathbf{A} = \mathbf{A}_c \odot \sigma(\mathbf{M}), \mathbf{X} = \mathbf{Z} + (\mathbf{X}_c - \mathbf{Z}) \odot \mathbf{F})$$

  - n: number of nodes
  - d: number of features
  - σ: sigmoid function
  - $\mathbf{A}_c$: adjacency matrix of computation graph
  - $\mathbf{X}_c$: node feature matrix of computation graph
  - $\mathbf{M} \in \mathbb{R}^{n \times n}$, $\mathbf{F} \in \{0,1\}^{n \times d}$: mask matrices
  - $\mathbf{Z}$: random variable sampled from empirical distribution

[1] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J.. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.

Overview of Part IV

Beyond Fairness on Graphs

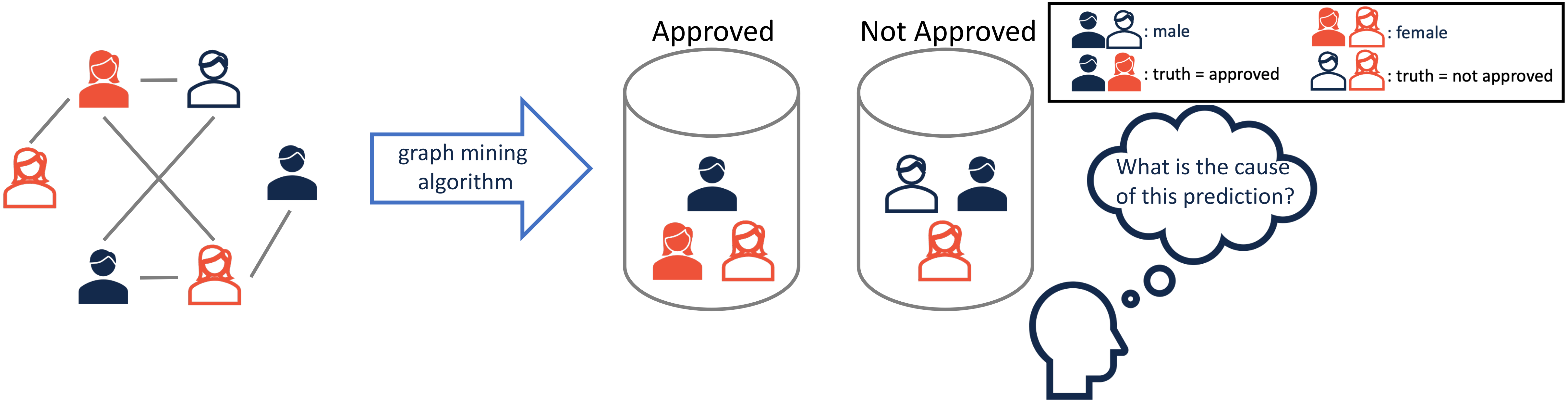
Explainability
- GNNExplainer

Accountability
- N2N

Robustness
- Nettack
- GNN-SVD

Related Problem #2: Accountability

- **Motivation: how** do mining results relate to graph topology?
- **Goal:** find influential elements w.r.t. the graph mining results
- **Example:** loan approval



[1] Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H.. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.
[2] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.

N2N: Formulation

- **N2N:** *network* $\mathbf{A}$ to *derivative network* $\mathbf{B}$
- **Intuition:** influential → high impact if perturbed
- **Edge influence:** derivative of $f(\mathbf{Y}^*)$ w.r.t. the edge

$$\mathbf{B}[i,j] = \frac{\mathrm{d}f(\mathbf{Y}^*)}{\mathrm{d}\mathbf{A}[i,j]}$$

- **Derivative network**

$$\mathbf{B} = \frac{\mathrm{d}f(\mathbf{Y}^*)}{\mathrm{d}\mathbf{A}} = \begin{cases} \frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}} + \left(\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}}\right)^T - \mathrm{diag}\left(\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}}\right), & \text{if undirected} \\ \boxed{\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}}}, & \text{if directed} \end{cases}$$

$$\mathrm{s.t.} \quad \mathbf{Y}^* = \mathrm{argmin}_Y L(\mathbf{A}, \mathbf{Y}, \theta)$$

key component to calculate

- **Question:** how to efficiently calculate the partial derivative?

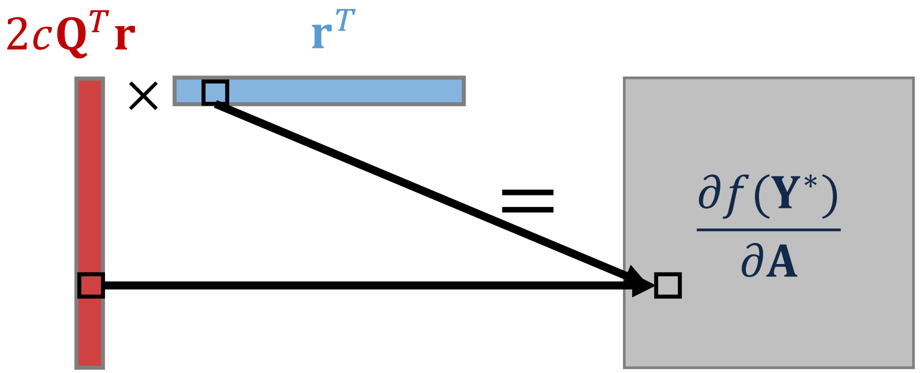
[1] Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H.. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.
[2] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.

Instantiation #1: PageRank

- **Basics of PageRank**
  - **Goal:** importance of nodes = probability a random walker land on the nodes
  - **Mining results:** $\mathbf{Y}^* = \mathbf{r} = (1-c)\mathbf{Q}\mathbf{e}$
    - $\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$
- **N2N for PageRank**
  - $f()$ **function:** $f(\mathbf{Y}^*) = \|\mathbf{r}\|_2^2$
  - **Partial derivative**

$$\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}} = 2c\mathbf{Q}^T\mathbf{r}\mathbf{r}^T$$

  - **Time Complexity:** $O(m)$; **space complexity:** $O(m+n)$
    - m = number of edges
    - n = number of nodes
  - **Remark**
    - N2N for PageRank is submodular



[1] Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H.. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.

Instantiation #2: HITS

- **Basics of HITS**
  - **Goal:** importance of nodes = (hub scores $\mathbf{u}$, authority scores $\mathbf{v}$)
  - **Mining results:** solve by rank-1 SVD
    - $\mathbf{u}$ = first left singular vector of $\mathbf{A}$ = principal eigenvector of $\mathbf{A}\mathbf{A}^T$
    - $\mathbf{v}$ = first right singular vector of $\mathbf{A}$ = principal eigenvector of $\mathbf{A}^T\mathbf{A}$
- **N2N for HITS**
  - ***f*() function:** $f(\mathbf{Y}^*) = \lambda_1 - \lambda_2$
    - λ_1 and λ_2 are the first and second largest eigenvalue of $\mathbf{A}^T\mathbf{A}$
  - **Partial derivative**

rank-2 SVD on $\mathbf{A}$

$$\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}} = 2\left(\mathbf{u_1}\,\mathbf{u_1}^T\mathbf{A} - \mathbf{u_2}\,\mathbf{u_2}^T\mathbf{A}\right) = 2\left(\boxed{\mathbf{u_1}\delta_1\mathbf{v_1}^T - \mathbf{u_2}\delta_2\mathbf{v_2}^T}\right)$$

- δ_1: largest singular value
- δ_2: second largest singular value

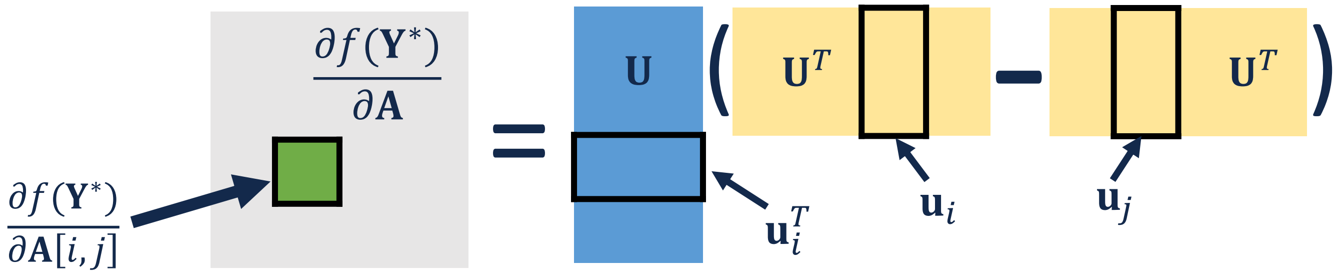
  - **Time Complexity:** $O(m+n)$; **space complexity:** $O(m+n)$
    - m = number of edges
    - n = number of nodes

Instantiation #3: Spectral Clustering

- **Basics of spectral clustering**
  - **Goal:** find k clusters such that $\begin{cases} \text{maximize intra-connectivity} \\ \text{minimize inter-connectivity} \end{cases}$
  - **Mining results:** $\mathbf{Y}^* = \mathbf{U} =$ eigenvectors of with k smallest eigenvalues
- **N2N for spectral clustering**
  - $\boldsymbol{f}()$ **function:** $f(\mathbf{Y}^*) = \mathrm{Tr}(\mathbf{U}^T\mathbf{L}\mathbf{U})$
  - **Partial derivative**

$$\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}} = \mathrm{diag}(\mathbf{U}\mathbf{U}^T)\mathbf{1}_{n\times n} - \mathbf{U}\mathbf{U}^T$$

  - **Time Complexity:** $O(k(m+n) + k^2 n)$; **space complexity:** $O(kn+m)$
    - m = number of edges
    - n = number of nodes
    - k = number of clusters



[1] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.

Instantiation #4: Matrix Completion

- **Basics of matrix completion**
  - **Goal:** learn low-rank matrices for n_1 users and n_2 items
  - **Optimization problem**

$$\min_{\mathbf{U},\mathbf{V}} \quad \|\mathrm{proj}_{\Omega}(\mathbf{A} - \mathbf{U}\mathbf{V}^T)\|_F^2 + \lambda_u \|\mathbf{U}\|_F^2 + \lambda_v \|\mathbf{V}\|_F^2$$

    - Ω = {observations}, λ_u, λ_u for regularization
- **N2N for matrix completion**
  - $\boldsymbol{f}()$ **function:** $f(\mathbf{Y}^*) = \|\mathbf{U}\mathbf{V}^T\|_F^2$
  - **Element-wise solution**

$$\frac{\partial f(\mathbf{Y}^*)}{\partial \mathbf{A}[i,j]} = 2\mathbf{U}[i,:]\mathbf{V}^T\mathbf{V}\mathbf{C}_i^{-1}\mathbf{V}[j,:]^T + 2\mathbf{V}[j,:]\mathbf{U}^T\mathbf{U}\mathbf{D}_j^{-1}\mathbf{U}[i,:]^T$$

    - Given mining results $\mathbf{U}$ and $\mathbf{V}$, precompute $\mathbf{U}^T\mathbf{U}$, $\mathbf{V}^T\mathbf{V}$, $\mathbf{C}_i$ and $\mathbf{D}_j$ during optimization
  - **Amortized time complexity:** $O(k^3(n_1 + n_2) + k^2 m)$; **space complexity:** $O(k^2(n_1 + n_2) + m)$
    - m = number of edges
    - n_1 = number of users
    - n_2 = number of items
    - k = dimension of latent factors

[1] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.

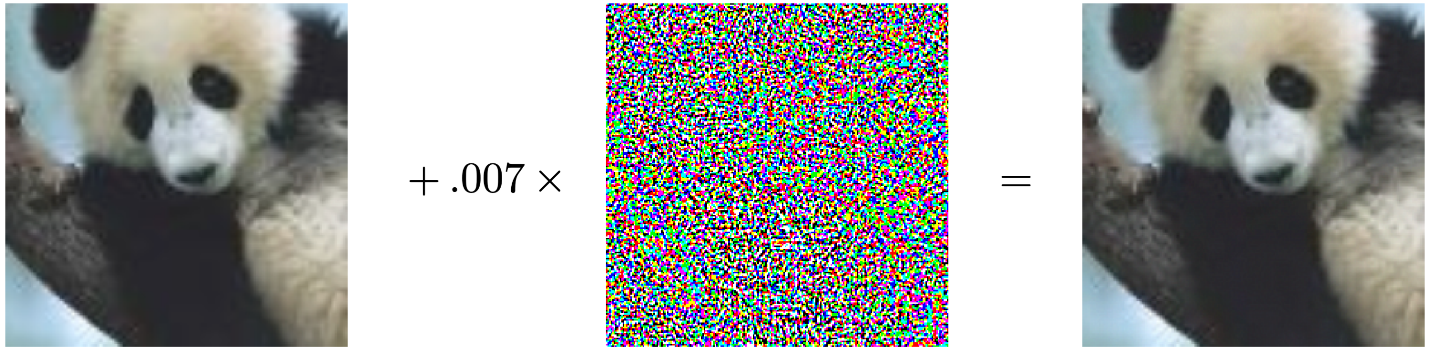
Overview of Part IV

Beyond Fairness on Graphs

- **Explainability**
  - GNNExplainer
- **Accountability**
  - N2N
- **Robustness**
  - Nettack
  - GNN-SVD

Related Problem #3: Robustness

- **Observation:** neural networks are sensitive to random perturbation



$\boldsymbol{x}$ — "panda" 57.7% confidence

$\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ — "nematode" 8.2% confidence

$\boldsymbol{x} + \epsilon \mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ — "gibbon" 99.3 % confidence

  – GNN, as a type of neural networks, makes no exception

- **Questions**
  – How to attack GNN so it makes bad predictions?
  – How to defend against such adversarial attacks?

[1] Goodfellow, I. J., Shlens, J., & Szegedy, C.. Explaining and Harnessing Adversarial Examples. ICLR 2015.
[2] Zügner, D., Akbarnejad, A., & Günnemann, S.. Adversarial Attacks on Neural Networks for Graph Data. KDD 2018.
[3] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., & Papalexakis, E. E.. All You Need is Low (Rank): Defending Against Adversarial Attacks on Graphs. WSDM 2020.

Attacking GNN: Nettack

- **Goal:** attack GNN with unnoticeable perturbation on graph and features
- **Optimization problem**

$$\underset{\mathbf{A}',\mathbf{X}'}{\operatorname{argmax}} \ \max_{c=c_{\text{old}}} \boxed{\log \mathbf{Z}[u,c] - \log \mathbf{Z}[u,c_{\text{old}}]}$$

Maximize the classification loss of a target node

$$\text{s.t.} \quad \boxed{\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\mathbf{A}',\mathbf{X}',\boldsymbol{\theta}) \\ \mathbf{Z} &= \mathrm{GCN}(\mathbf{A}',\mathbf{X}',\boldsymbol{\theta}^*) \end{aligned}}$$

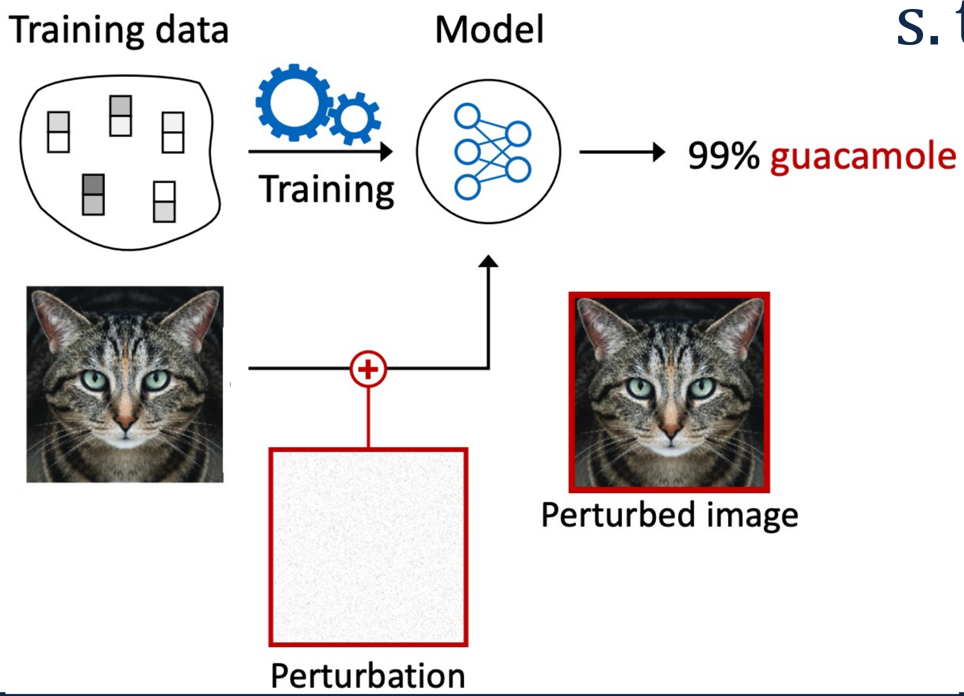
Measure impact after GCN re-training
* For evasion attack, $\boldsymbol{\theta}$ is fixed

$$\boxed{(\mathbf{A}',\mathbf{X}') \approx (\mathbf{A},\mathbf{X})}$$

Unnoticeable attack

$$\boxed{\|\mathbf{A}-\mathbf{A}'\|_0 + \|\mathbf{X}-\mathbf{X}'\|_0 \le \Delta}$$

Budgeted attack



[1] Zügner, D., Akbarnejad, A., & Günnemann, S.. Adversarial Attacks on Neural Networks for Graph Data. KDD 2018.

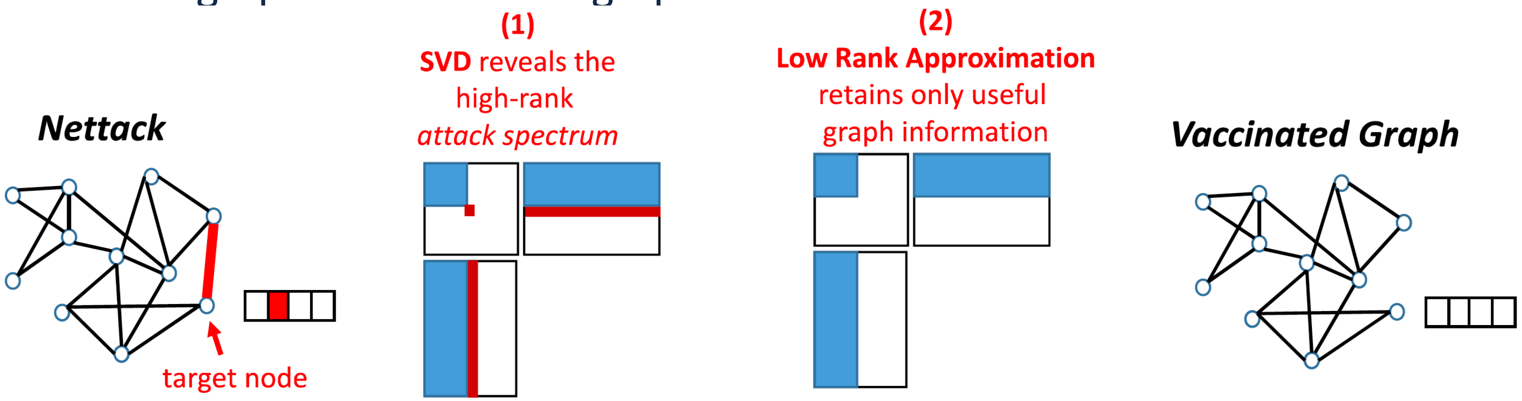
Overview of Part IV

Beyond Fairness on Graphs

- **Explainability**
  - GNNExplainer
- **Accountability**
  - N2N
- **Robustness**
  - Nettack
  - GNN-SVD

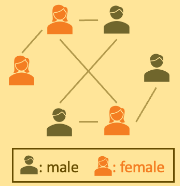
Defending GNN: GNN-SVD

- **Motivation:** GNN is vulnerable to adversarial attack
  - How to make GNN more robust?
- **Observation:** Nettack is a high-rank attack
  - High-rank spectrum (i.e., small singular values) will change after attack
- **Key idea:** low-rank approximation can resist such attack
- **Steps**
  - Take a truncated SVD of the input graph structure
  - Reconstruct the graph with top-k singular values and their singular vectors
  - Output the reconstructed graph as vaccinated graph

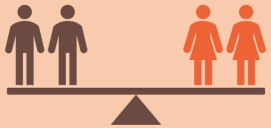


[1] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., & Papalexakis, E. E.. All You Need is Low (Rank): Defending Against Adversarial Attacks on Graphs. WSDM 2020.

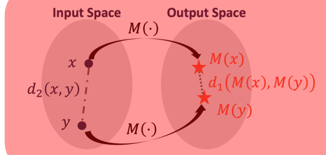
Roadmap



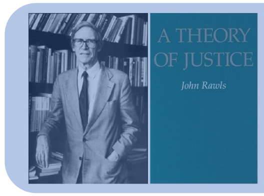
Introduction



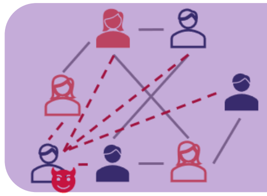
Part I: Group Fairness on Graphs



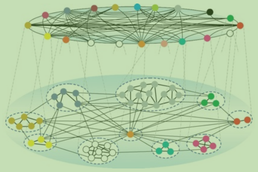
Part II: Individual Fairness on Graphs



Part III: Other Fairness on Graphs



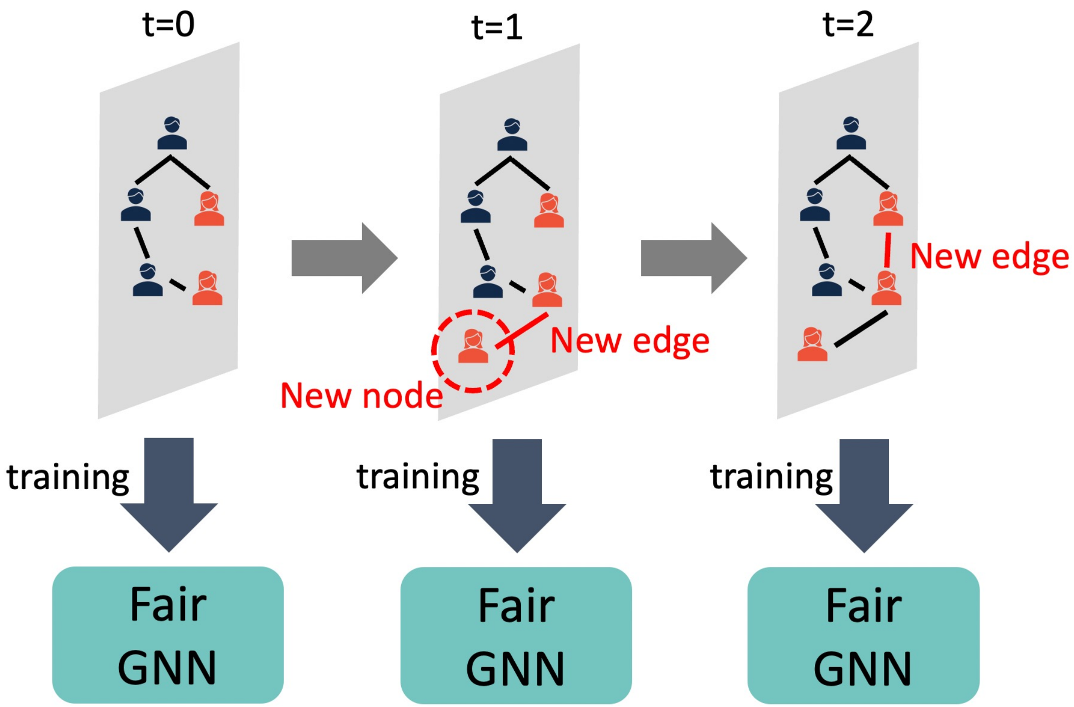
Part IV: Beyond Fairness on Graphs



Part V: Future Trends

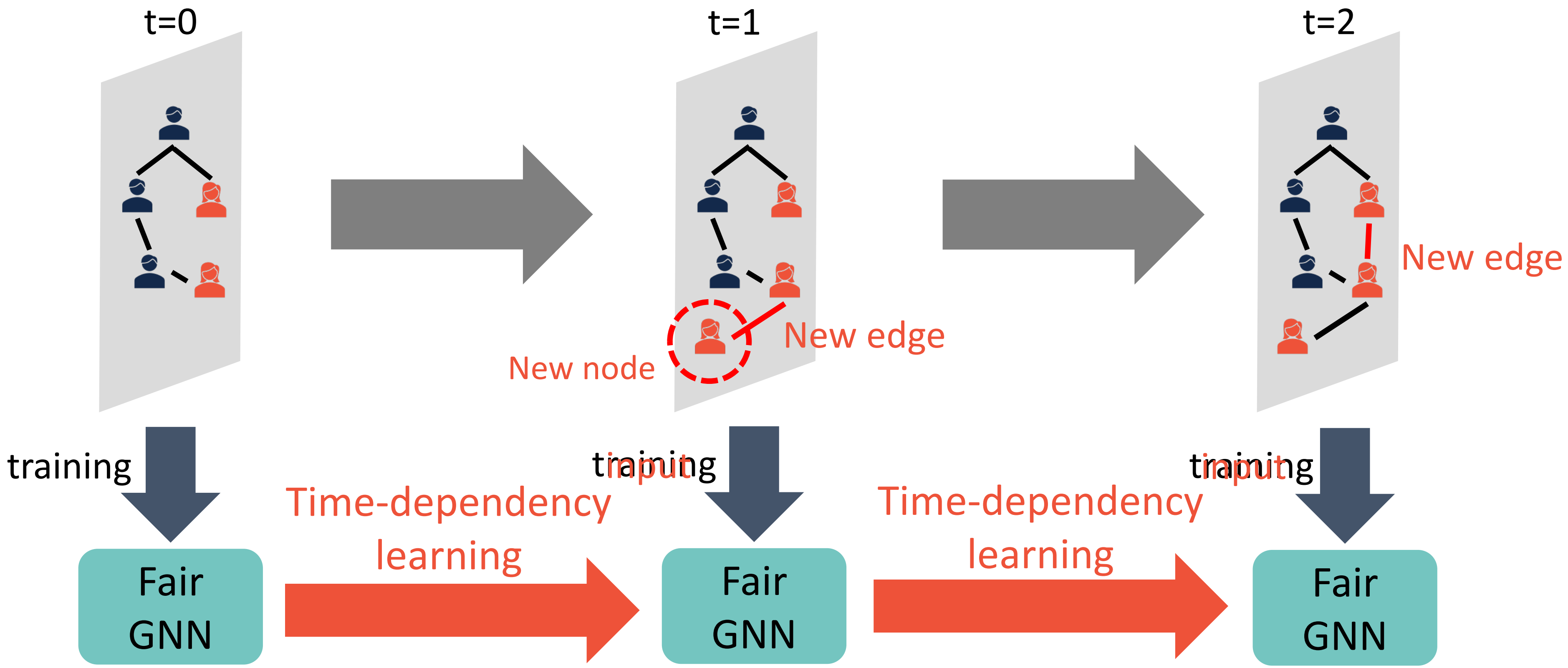
Fairness on Dynamic Graphs

- **Motivation:** networks are dynamically changing over time
  - **New nodes:** new accounts on social network platforms (e.g., Facebook, Twitter)
  - **New edges:** new engagements among people on social networks (e.g., follow, retweet)
- **Trivial solution:** re-run the fair graph mining algorithm from scratch at each timestamp
- **Limitations**
  - Time-consuming to re-train the mining model
  - Fail to capture the dynamic fairness-related information
- **Questions**
  - How to efficiently update the mining results and ensure the fairness at each timestamp?
  - How to characterize the impact of dynamics over the bias measure?



Fairness on Dynamic Graphs

- **Possible method:** fair graph mining model with time-dependency learning module
  - **Efficient update:** dynamic tracking module
  - **Temporal information learning:** gated recurrent unit (GRU)

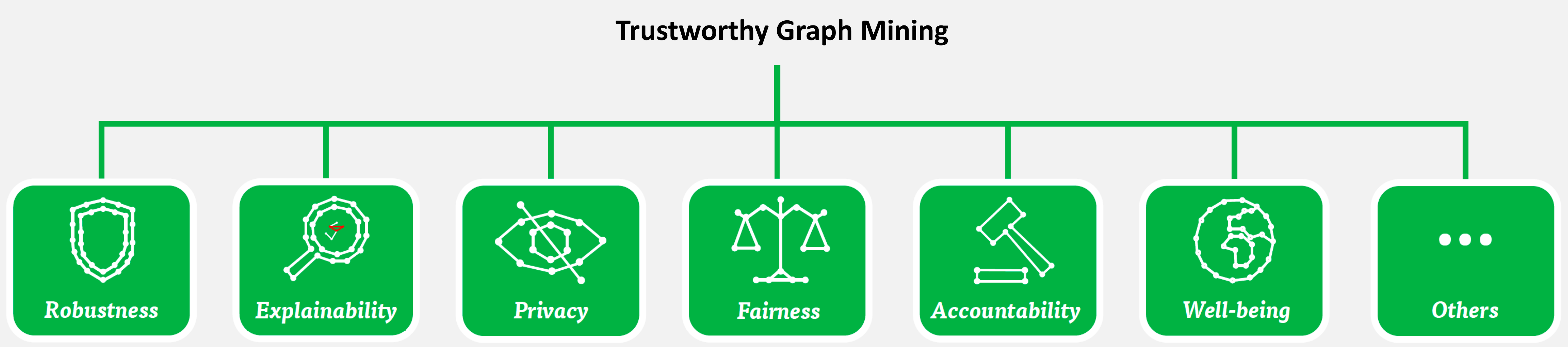


Benchmark and Evaluation Metrics

- **Motivation:** there is no consensus on the experimental settings for fair graph mining
  - Which graph(s) we should use for fair graph mining?
  - What could be the sensitive attribute(s) for each dataset to be used?
  - What should be the evaluation metric for each type of fairness on graphs?
  - How to split the dataset for training, validation and test?
- **Consequences**
  - Different settings for different research works
  - Hardly fair comparison among debiasing methods
- **Call:** the community should work together toward
  - A consensus on the experimental settings
  - A benchmark for fair comparison of different methods

Fairness vs. Other Social Aspects

- **Overview:** trustworthy graph mining

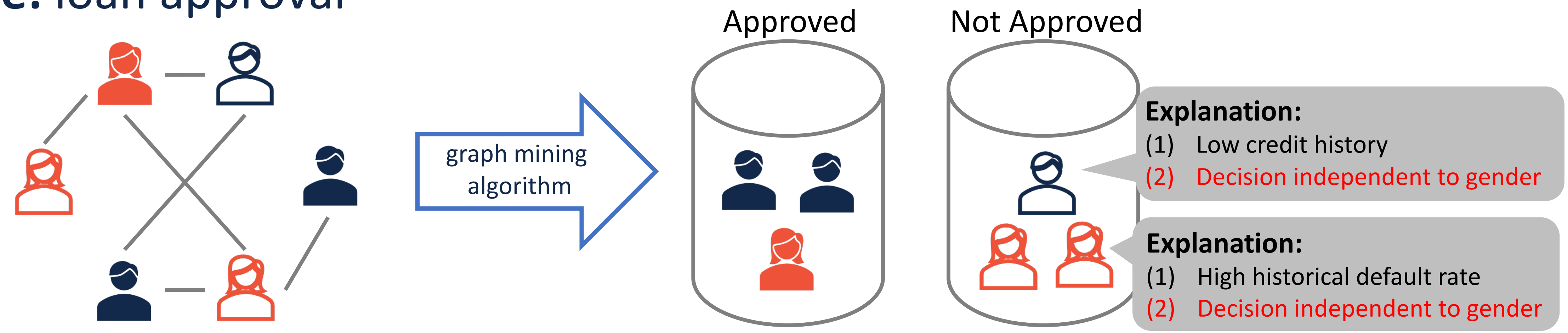


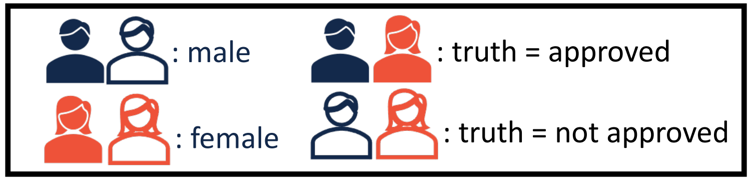
- **Motivation:** tensions among the social aspects
- **Fairness vs. privacy**
  - Is fairness related to privacy preservation on graphs?
  - Will preserving privacy help ensuring fairness, or vice versa?

[1] Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., & Pei, J.. Trustworthy Graph Neural Networks: Aspects, Methods and Trends. arXiv.
[2] Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., ... & Wang, S.. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. arXiv.

Fairness vs. Explainability

- **Research questions**
  - Are the existing debiasing methods transparent?
  - If not, can we open the black box of debiasing methods on graphs?
- **Example:** loan approval



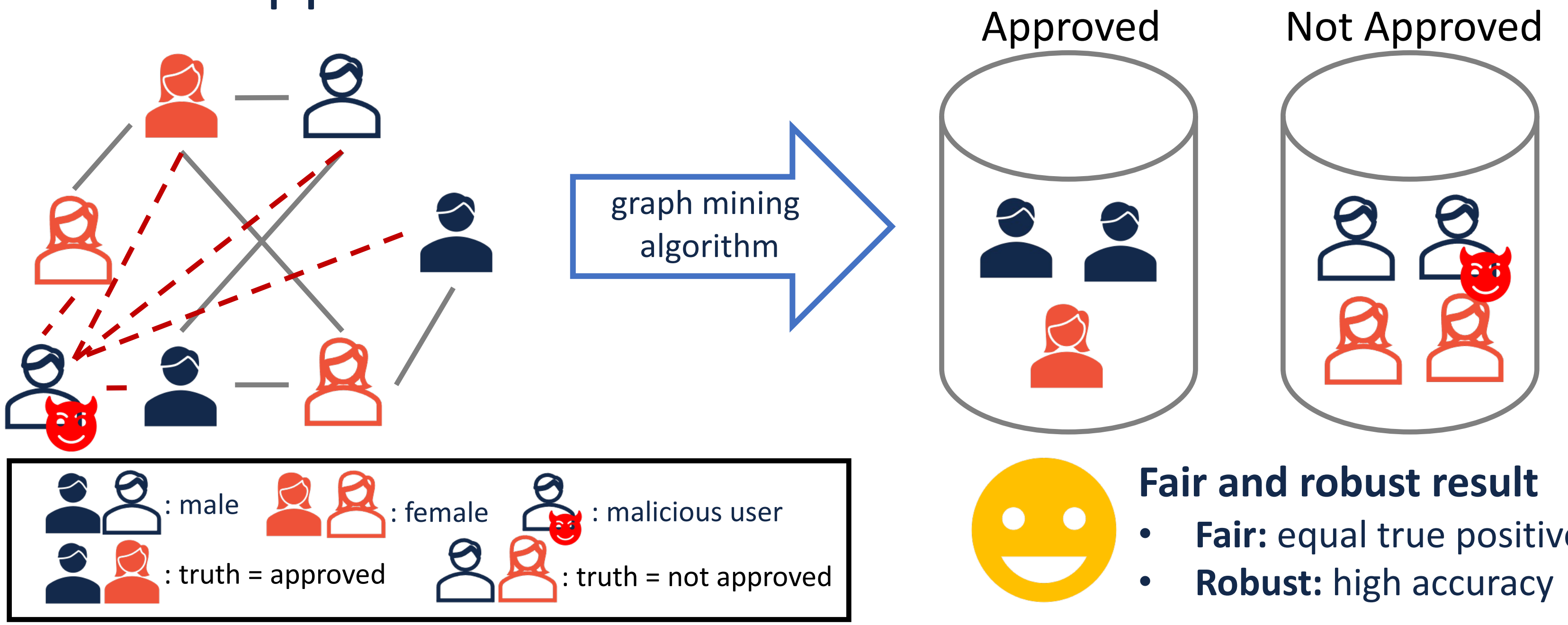


Fair and transparent result

- **Fair:** equal true positive rate
- **Transparent:** explanation on the usage of sensitive information

Fairness vs. Robustness

- **Research questions**
  - Will existing adversarial attack strategies affect the fairness of mining model?
  - Are the existing debiasing methods robust against random noise and adversary?
- **Example:** loan approval



Fair and robust result

- **Fair:** equal true positive rate
- **Robust:** high accuracy

Takeaways

- **Introduction to algorithmic fairness on graphs**
  - Background, challenges, related problems
- **Group fairness on graphs**
  - Classic graph mining: ranking, clustering
  - Advanced graph mining: node embedding, graph neural networks
- **Individual fairness on graphs**
  - Laplacian regularization-based method, ranking-based method
- **Other fairness on graphs**
  - Counterfactual fairness, degree fairness
- **Beyond fairness on graphs**
  - Explainability, accountability, robustness
- **Future directions**
  - Fairness on dynamic graphs
  - Benchmark and evaluation metrics for algorithmic fairness on graphs
  - Interplay between fairness and other aspects of trustworthiness

Resources

- **Datasets:** https://github.com/yushundong/Graph-Mining-Fairness-Data
- **Surveys**
  - Zhang, W., Weiss, J. C., Zhou, S., & Walsh, T.. Fairness Amidst Non-IID Graph Data: A Literature Review. arXiv preprint arXiv:2202.07170.
  - Dong, Y., Ma, J., Chen, C., & Li, J.. Fairness in Graph Mining: A Survey. arXiv preprint arXiv:2204.09888.
  - Zhang, H., Wu, B., Yuan, X., Pan, S., Tong, H., & Pei, J.. Trustworthy Graph Neural Networks: Aspects, Methods and Trends. arXiv preprint arXiv:2205.07424.
  - Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., ... & Wang, S.. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. arXiv preprint arXiv:2204.08570.
- **Related tutorials**
  - Fair Graph Mining
    - http://jiank2.web.illinois.edu/tutorial/cikm21/fair_graph_mining.html
  - Fairness in Networks
    - https://algofairness.github.io/kdd-2021-network-fairness-tutorial/

Acknowledgements

- Part of the slides are credited to the following authors
  (in alphabetic order of last name)
  - Chirag Agarwal (Harvard University)
  - Avishek Joey Bose (McGill University)
  - Yushun Dong (University of Virginia)
  - Matthäus Kleindessner (Amazon)
  - Peizhao Li (Brandeis University)
  - Jing Ma (University of Virginia)
  - Evaggelia Pitoura (University of Ioannina)
  - Xianfeng Tang (Amazon)
  - Panayiotis Tsaparas (University of Ioannina)
- If you would like to re-use these slides, please contact the original authors.

References

- Tsioutsiouliklis, S., Pitoura, E., Tsaparas, P., Kleftakis, I., & Mamoulis, N.. Fairness-Aware PageRank. WWW 2021.
- Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., & Tsaparas, P.. Link Recommendations for PageRank Fairness. WWW 2022.
- Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.
- Bose, A., & Hamilton, W.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.
- Rahman, T., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.
- Khajehnejad, A., Khajehnejad, M., Babaei, M., Gummadi, K. P., Weller, A., & Mirzasoleiman, B.. CrossWalk: Fairness-enhanced Node Representation Learning. AAAI 2022.
- Dai, E., & Wang, S.. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. WSDM 2021.

References

- Kang, J., He, J., Maciejewski, R., & Tong, H.. InFoRM: Individual Fairness on Graph Mining. KDD 2020.
- Dong, Y., Kang, J., Tong, H., & Li, J.. Individual Fairness for Graph Neural Networks: A Ranking based Approach. KDD 2021
- Agarwal, C., Lakkaraju, H., & Zitnik, M.. Towards a Unified Framework for Fair and Stable Graph Representation Learning. UAI 2021.
- Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., & Li, J.. Learning Fair Node Representations with Graph Counterfactual Fairness. WSDM 2022.
- Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., ... & Wang, S.. Investigating and Mitigating Degree-related Biases in Graph Convolutional Networks. CIKM 2020.
- Kang, J., Zhu, Y., Xia, Y., Luo, J., & Tong, H.. RawlsGCN: Towards Rawlsian Difference Principle on Graph Convolutional Network. WWW 2022.

References

- Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J.. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.
- Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H.. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.
- Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.
- Zügner, D., Akbarnejad, A., & Günnemann, S.. Adversarial Attacks on Neural Networks for Graph Data. KDD 2018.
- Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., & Papalexakis, E. E.. All You Need is Low (Rank): Defending Against Adversarial Attacks on Graphs. WSDM 2020.