InFoRM: Individual Fairness on Graph Mining

Jian Kang  Jingrui He  Ross Maciejewski  Hanghang Tong
Graph Mining: Applications

Graph Mining: How To

• **Graph Mining Pipeline**
  - Input graph
  - Mining model
  - Mining results

• **Example:** job application classification

• **Question:** are the mining results fair or biased?

- (male): 50%
- (female): 50%

- (male): ?%
- (female): ?%

- (male): ?%
- (female): ?%
Algorithmic Fairness in Machine Learning

- **Goal**: minimize unintentional bias caused by machine learning algorithms

- **Existing Measures**
  - Group fairness
    - Disparate impact [1]
    - Statistical parity [2]
    - Equal odds [3]
  - Counterfactual fairness [4]
  - Individual fairness [5]

Group Fairness: Statistical Parity

• **Definition:** candidates in protected and unprotected groups have equal probability of being assigned to a predicted class $c$
  \[
  \Pr_{+}(y = c) = \Pr_{-}(y = c)
  \]
  \[= \Pr_{+}(y = c): \text{probability of being assigned to } c \text{ for protected group; } \Pr_{-}(y = c) \text{ is for unprotected group}
  \]

• **Illustrative Example:** job application classification

• **Advantages:**
  – Intuitive and well-known
  – No impact of sensitive attributes

• **Disadvantage:** fairness can still be ensured when
  – Choose qualified candidates in one group
  – Choose candidates randomly in another group
Individual Fairness

- **Problem of Group Fairness:** different forms of bias in different settings
  - **Question:** which fairness notion should we apply?
- **Principle:** similar individuals should receive similar algorithmic outcomes [1]
  - **Rooted in definition of fairness [2]:** lack of favoritism from one side or another
- **Definition:** given two distance metrics $d_1$ and $d_2$, a mapping $M$ satisfies individual fairness if for every $x, y$ in a collection of data $D$
  $$d_1(M(x), M(y)) \leq d_2(x, y)$$
- **Illustrative Example:**

- **Advantage:** finer granularity than group fairness
- **Disadvantage:** hard to find proper distance metrics

---

Algorithmic Fairness in Machine Learning

• **Goal:** minimize unintentional discrimination caused by machine learning algorithms

• **Existing Measures**
  – Group fairness
    • Disparate impact [1]
    • Statistical parity [2]
    • Equal odds [3]
  – Counterfactual fairness [4]
  – Individual fairness [5]

• **Limitation:** IID assumption in traditional machine learning
  – Might be violated by the non-IID nature of graph data

Algorithmic Fairness in Graph Mining

- **Fair Spectral Clustering** [1]
  - **Fairness notion:** disparate impact

- **Fair Graph Embedding**
  - Fairwalk [2], compositional fairness constraints [3]
    - **Fairness notion:** statistical parity
  - MONET [4]
    - **Fairness notion:** orthogonality of metadata and graph embedding

- **Fair Recommendation**
  - Information neural recommendation [5]
    - **Fairness notion:** statistical parity
  - Fairness for collaborative filtering [6]
    - **Fairness notion:** four metrics that measure the differences in estimation error between ground-truth and predictions across protected and unprotected groups

---

Compositional Fairness Constraints for Graph Embeddings [1]

- **Goal:** learn graph embeddings that is fair w.r.t. a combination of different sensitive attributes
- **Fairness definition:** mutual information between sensitive attributes and embedding is 0
  - Imply statistical parity
- **Method:** adversarial training
  - **Key idea:** train filters for each sensitive attribute so that embeddings fail to predict this attribute

Algorithmic Fairness in Graph Mining

• **Fair Spectral Clustering** [1]
  – **Fairness notion:** disparate impact

• **Fair Graph Embedding**
  – **Fairwalk** [2], compositional fairness constraints [3]
    • **Fairness notion:** statistical parity
  – **MONET** [4]
    • **Fairness notion:** orthogonality of metadata and graph embedding

• **Fair Recommendation**
  – Information neural recommendation [5]
    • **Fairness notion:** statistical parity
  – Fairness for collaborative filtering [6]
    • **Fairness notion:** four metrics that measure the differences in estimation error between ground-truth and predictions across protected and unprotected groups

• **Observation:** all of them focus on group-based fairness!

InFoRM: Individual Fairness on Graph Mining

• Research Questions

Q1. Measures: how to quantitatively measure individual bias?
Q2. Algorithms: how to enforce individual fairness?
Q3. Cost: what is the cost of individual fairness?
Graph Mining Algorithms

- **Graph Mining: An Optimization Perspective**

  - **Input:**
    - Input graph $A$
    - Model parameters $\theta$
  
  - **Output:** mining results $Y$

  - **Examples:** ranking vectors, class probabilities, embeddings

  - Minimize loss function $l(A, Y, \theta)$
### Classic Graph Mining Algorithms

#### Examples of Classic Graph Mining Algorithm

<table>
<thead>
<tr>
<th>Mining Task</th>
<th>Loss Function $L()$</th>
<th>Mining Result $Y^*$</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>$\min_r cr'(I - A)r + (1 - c)|r - e|^2_F$</td>
<td>PageRank vector $r$</td>
<td>damping factor $c$</td>
</tr>
<tr>
<td>Spectral Clustering</td>
<td>$\min_u \text{Tr} (U'LU)$ s.t. $U'U = I$</td>
<td>eigenvectors $U$</td>
<td># clusters $k$</td>
</tr>
<tr>
<td>LINE (1st)</td>
<td>$\min_x \sum_{i=1}^{n} \sum_{j=1}^{n} A[i, j] (\log g(\langle X[j,:]; X[i,:]\rangle))$ $+ b \sum_{j' \sim p_n} \log g(\langle X[j',:]; X[i,:]\rangle)$</td>
<td>embedding matrix $X$</td>
<td>embedding dimension $d$</td>
</tr>
</tbody>
</table>

#### Diagrams

- **PageRank**
  - Ranking algorithm

- **Spectral Clustering**
  - Eigenvectors

- **LINE (1st)**
  - Embedding matrix

- **Network**
  - Adjacency matrix

- **Vector based representation**
  - Low dimensional space
Roadmap

• Motivations ✓
• InFoRM Measures
• InFoRM Algorithms
  – Debiasing the Input Graph
  – Debiasing the Mining Model
  – Debiasing the Mining Results
• InFoRM Cost
• Experimental Results
• Conclusions
Problem Definition: InFoRM Measures

• Questions
  – How to determine if the mining results are fair?
  – How to quantitatively measure the overall bias?

• Input
  – Node-node similarity matrix $S$
    • Non-negative, symmetric
  – Graph mining algorithm $l(A, Y, \theta)$
    • Loss function $l(\cdot)$
    • Additional set of parameters $\theta$
  – Fairness tolerance parameter $\epsilon$

• Output
  – Binary decision on whether the mining results are fair
  – Individual bias measure $\text{Bias}(Y, S)$
Measuring Individual Bias: Formulation

- **Principle:** similar nodes $\rightarrow$ similar mining results

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

  - **Intuition:** if $S[i,j]$ is high, $\frac{\epsilon}{S[i,j]}$ is small $\Rightarrow$ push $\mathbf{Y}[i, :]$ and $\mathbf{Y}[j, :]$ to be more similar
  - **Observation:** Inequality should hold for every pairs of nodes $i$ and $j$
    - **Problem:** too restrictive to be fulfilled

- **Relaxed Criteria:**
  \[
  \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{Y}[i, :] - \mathbf{Y}[j, :]\|_F^2 S[i,j] = 2\text{Tr}(\mathbf{Y}'\mathbf{L}\mathbf{S}\mathbf{Y}) \leq m\epsilon = \delta
  \]
Measuring Individual Bias: Solution

• InFoRM (Individual Fairness on Graph Mining)
  – Given (1) a graph mining results $Y$, (2) a symmetric similarity matrix $S$ and (3) a constant fairness tolerance $\delta$
  – $Y$ is individually fair w.r.t. $S$ if it satisfies
    \[ \text{Tr}(Y'LSY) \leq \frac{\delta}{2} \]
  – Overall individual bias is $\text{Bias}(Y, S) = \text{Tr}(Y'LSY)$

Lipschitz Property of Individual Fairness

• Connection to Lipschitz Property
  – $(D_1, D_2)$-Lipschitz property [1]: a function $f$ is $(D_1, D_2)$-Lipschitz if it satisfies
    $$D_1(f(i), f(j)) \leq LD_2(i, j), \forall (x, y)$$
    • $L$ is Lipschitz constant
  – InFoRM naturally satisfies $(D_1, D_2)$-Lipschitz property as long as
    • $f(i) = Y[i,:]$
    • $D_1(f(i), f(j)) = ||Y[i,:] - Y[j,:]||^2_2, D_2(i, j) = \frac{1}{s[i,j]}$
  – Lipschitz constant of InFoRM is $\epsilon$
Roadmap

• Motivations ✓
• InFoRM Measures ✓
• InFoRM Algorithms
  – Debiasing the Input Graph
  – Debiasing the Mining Model
  – Debiasing the Mining Results
• InFoRM Cost
• Experimental Results
• Conclusions
Problem Definition: InFoRM Algorithms

• **Question**: how to mitigate the bias of the mining results?

• **Input**
  – Node-node similarity matrix $S$
  – Graph mining algorithm $l(A, Y, \theta)$
  – Individual bias measure $Bias(Y, S)$
    • Defined in the previous problem (InFoRM Measures)

• **Output**: a revised mining results $Y^*$ that minimizes
  – Loss function $l(A, Y, \theta)$
  – Individual bias measure $Bias(Y, S)$
Mitigating Individual Bias: How To

• **Graph Mining Pipeline**
  - input graph $A$
  - mining model w/ parameter $\theta$
  - mining results $Y$
  - minimize $l(A, Y, \theta)$

• **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
Debiasing the Input Graph

• **Goal:** bias mitigation via a pre-processing strategy

• **Intuition:** learn a new topology of graph $\tilde{A}$ such that
  – $\tilde{A}$ is as similar to the original graph $A$ as possible
  – Bias of mining results on $\tilde{A}$ is minimized

• **Optimization Problem**

$$\min_Y J = \|\tilde{A} - A\|_F^2 + \alpha \text{Tr}(Y' L_S Y)$$

subject to $Y = \arg\min_Y l(\tilde{A}, Y, \theta)$

• **Challenge:** bi-level optimization
  – **Solution:** exploration of KKT conditions [1, 2]

---

Debiasing the Input Graph

• Considering the KKT conditions,

\[
\min_Y J = \|\tilde{A} - A\|_F^2 + \alpha \text{Tr}(Y' L_S Y)
\]

s. t. \( \partial_Y l(\tilde{A}, Y, \theta) = 0 \)

• **Proposed Method**
  
  (1) Fix \( \tilde{A} \) (\( \tilde{A} = A \) at initialization), find \( Y \) using current \( \tilde{A} \)
  
  (2) Fix \( Y \), update \( \tilde{A} \) by gradient descent
  
  (3) Iterate between (1) and (2)

• **Problem:** how to calculate gradient w.r.t. \( \tilde{A} \)?
Debiasing the Input Graph

• Calculating Gradient

\[
\frac{\partial J}{\partial A} = 2(\tilde{A} - A) + \alpha \left[ \text{Tr} \left( 2\tilde{Y}L_s \frac{\partial \tilde{Y}}{\partial A[i,j]} \right) \right]
\]

\[
\frac{dJ}{dA} = \begin{cases} 
\frac{\partial J}{\partial A} + \left( \frac{\partial J}{\partial A} \right)' - \text{diag} \left( \frac{\partial J}{\partial A} \right), & \text{if undirected} \\
\frac{\partial J}{\partial A}', & \text{if directed}
\end{cases}
\]

- \( \tilde{Y} \) satisfies \( \partial_Y l(\tilde{A}, Y, \theta) = 0 \)
- \( H = \left[ \text{Tr} \left( 2\tilde{Y}L_s \frac{\partial \tilde{Y}}{\partial A[i,j]} \right) \right] \) is a matrix with \( H[i,j] = \text{Tr} \left( 2\tilde{Y}L_s \frac{\partial \tilde{Y}}{\partial A[i,j]} \right) \)

• Question: how to efficiently calculate \( H \)?
Instantiation #1: PageRank

- **Goal:** efficiently calculate $H$ for PageRank
- **Mining Results $Y$:** $r = (1 - c)Qe$
- **Partial Derivatives $H$:** $H = 2cQ'L_Sr'r'$
- **Remarks:** $Q = (I - cA)^{-1}$
- **Time Complexity**
  - Straightforward: $O(n^3)$
  - Ours: $O(m_1 + m_2 + n)$
    - $m_A$: number of edges in $A$
    - $m_S$: number of edges in $S$
    - $n$: number of nodes
Instantiation #2: Spectral Clustering

- **Goal:** efficiently calculate $H$ for spectral clustering
- **Mining Results $Y$:** $U = \text{eigenvectors with } k \text{ smallest eigenvalues}$
- **Partial Derivatives $H$:**
  \[ H = 2 \sum_{i=1}^{k} (\text{diag}(M_i L_S u_i u_i')) 1_{n \times n} - M_i L_S u_i u_i' \]
- **Remarks:** $(\lambda_i, u_i) = i$-th smallest eigenpair, $M_i = (\lambda_i I - L_A)^+$
- **Time Complexity**
  - Straightforward: $O(k^2 (m + n) + k^3 n + kn^3)$
  - Ours: $O(k^2 (m + n) + k^3 n)$
Instantiation #3: LINE (1st)

- **Goal:** efficiently calculate $H$ for LINE (1st)

- **Mining Results $Y$:** $Y[i, :]Y[j, :]' = \log \frac{T(\tilde{A}[i,j]+\tilde{A}[j,i])}{d_id_j^{3/4}+d_i^{3/4}d_j} - \log b$
  - $d_i =$ outdegree of node $i$, $T = \sum_{i=1}^{n} d_i^{3/4}$ and $b =$ number of negative samples

- **Partial Derivatives $H$:** $H = 2f(\tilde{A} + \tilde{A}') \circ L_S - 2\text{diag}(BL_S)1_{n \times n}$

- **Remarks**
  - $f()$ calculates Hadamard inverse, $\circ$ calculates Hadamard product
  - $B = \frac{3}{4}f \left( d^{5/4}(d^{-1/4})' + \mathbf{d1}_{n \times n} \right) + f \left( d^{3/4}(d^{1/4})' + \mathbf{d1}_{n \times n} \right)$ with $d^x[i] = d_i^x$

- **Time Complexity**
  - Straightforward: $O(n^3)$
  - Ours: $O(m_1 + m_2 + n)$
    - $m_A$: number of edges in $A$
    - $m_S$: number of edges in $S$
    - $n$: number of nodes
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**
  \[
  \min_Y J = l(A, Y, \theta) + \alpha \text{Tr}(Y' L_S Y)
  \]

• **Solution**
  – **General:** solve by (stochastic) gradient descent
  \[
  \frac{\partial J}{\partial Y} = \frac{\partial l(A, Y, \theta)}{\partial Y} + 2\alpha L_S Y
  \]
  – **Task-specific:** solve by specific algorithm designed for the graph mining problem

• **Advantage**
  – Linear time complexity incurred in computing the gradient
Debiasing the Mining Model: Instantiations

- **PageRank**
  - **Objective Function:** \( \min_r c r'(I - A)r + (1 - c)\|r - e\|^2_F + \alpha r'L_S r \)
  - **Solution:** \( r^* = c \left( A - \frac{\alpha}{c} L_S \right) r^* + (1 - c)e \)
    - PageRank on new transition matrix \( A - \frac{\alpha}{c} L_S \)
    - If \( L_S = I - S \), then \( r^* = \left( \frac{c}{1+\alpha} A + \frac{\alpha}{1+\alpha} S \right) r^* + \frac{1-c}{1+\alpha} e \)

- **Spectral Clustering**
  - **Objective Function:** \( \min_U \text{Tr}(U' L_A U) + \alpha \text{Tr}(U' L_S U) = \text{Tr}(U' (L_A + \alpha S) U) \)
  - **Solution:** \( U^* = \) eigenvectors of \( L_A + \alpha S \) with \( k \) smallest eigenvalues
    - spectral clustering on an augmented graph \( A + \alpha S \)

- **LINE (1st)**
  - **Objective Function:** \( \max_{x_i, x_j} \log g(x_j x_i') + b \mathbb{E}_{j' \in P_n} \left[ \log g(-x_j, x_i') \right] - \alpha \|x_i - x_j\|^2_F S[i, j] \) \( \forall i, j = 1, \ldots, n \)
  - **Solution:** stochastic gradient descent
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**

\[
\min_Y J = \|Y - \bar{Y}\|_F^2 + \alpha \text{Tr}(Y'L_S Y)
\]

– \(\bar{Y}\) is the vanilla mining results

• **Solution:** \((I + \alpha S)Y^* = \bar{Y}\)

– convex loss function as long as \(\alpha \geq 0 \rightarrow\) global optima by \(\frac{\partial J}{\partial Y} = 0\)

– solve by conjugate gradient (or other linear system solvers)

• **Advantages**

– No knowledge needed on the input graph

– Model-agnostic
Roadmap

• Motivations ✓
• InFoRM Measures ✓
• InFoRM Algorithms ✓
  – Debiasing the Input Graph
  – Debiasing the Mining Model
  – Debiasing the Mining Results
• InFoRM Cost
• Experimental Results
• Conclusions
Problem Definition: InFoRM Cost

• **Question:** how to quantitatively characterize the cost of individual fairness?

• **Input**
  – Vanilla mining results $\overline{Y}$
  – Debiased mining results $Y^*$
    • Learned by the previous problem (InFoRM Algorithms)

• **Output:** an upper bound of $\|\overline{Y} - Y^*\|_F$

• **Debiasing Methods**
  – Debiasing the input graph
  – Debiasing the mining model
  – Debiasing the mining results

depend on specific graph topology/mining model

main focus of this paper
Cost of Debiasing the Mining Results

• Given
  – A graph with $n$ nodes and adjacency matrix $A$
  – A node-node similarity matrix $S$
  – Vanilla mining results $\bar{Y}$
  – Debiased mining results $Y^* = (I + \alpha S)^{-1}\bar{Y}$

• If $\|S - A\|_F = \delta$, we have
  $$\|\bar{Y} - Y^*\|_F \leq 2\sqrt{n} \left( \delta + \sqrt{\text{rank}(A)\sigma_{\max}(A)} \right) \|\bar{Y}\|_F$$

• Observation: the cost of debiasing the mining results depends on
  – The number of nodes $n$ (i.e. size of the input graph)
  – The difference $\delta$ between $A$ and $S$
  – The rank of $A$ could be small due to low-rank structures in real-world graphs
  – The largest singular value of $A$ could be small if $A$ is normalized
Cost of Debiasing the Mining Model: Case Study on PageRank

• Given
  – A graph with \( n \) nodes and symmetrically normalized adjacency matrix \( \mathbf{A} \)
  – A symmetrically normalized node-node similarity matrix \( \mathbf{S} \)
  – Vanilla PageRank vector \( \bar{\mathbf{r}} \)
  – Debiased PageRank vector \( \mathbf{r}^* = (\mathbf{I} + \alpha \mathbf{S})^{-1} \hat{\mathbf{Y}} \)

• If \( \| \mathbf{S} - \mathbf{A} \|_F = \delta \), we have
  \[
  \| \bar{\mathbf{r}} - \mathbf{r}^* \|_F \leq \frac{2\alpha n}{1 - c} \left( \delta + \sqrt{\text{rank}(\mathbf{A})} \sigma_{\text{max}}(\mathbf{A}) \right)
  \]

• Observation: the cost of debiasing PageRank depends on
  – The number of nodes \( n \) (i.e. size of the input graph)
  – The difference \( \delta \) between \( \mathbf{A} \) and \( \mathbf{S} \)
  – The rank of \( \mathbf{A} \) could be small due to low-rank structures in real-world graphs
  – The largest singular value of \( \mathbf{A} \) upper bounded by 1
Roadmap

• Motivations  
• InFoRM Measures  
• InFoRM Algorithms  
  – Debiasing the Input Graph  
  – Debiasing the Mining Model  
  – Debiasing the Mining Results  
• InFoRM Cost  
• Experimental Results  
• Conclusions
Experimental Settings

• Questions:
  RQ1. What is the impact of individual fairness in graph mining performance?
  RQ2. How effective are the debiasing methods?
  RQ3. How efficient are the debiasing methods?

• Datasets: 5 publicly available real-world datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>AstroPh</td>
<td>18,772</td>
<td>198,110</td>
</tr>
<tr>
<td>CondMat</td>
<td>23,133</td>
<td>93,497</td>
</tr>
<tr>
<td>Facebook</td>
<td>22,470</td>
<td>171,002</td>
</tr>
<tr>
<td>Twitter</td>
<td>7,126</td>
<td>35,324</td>
</tr>
<tr>
<td>PPI</td>
<td>3,890</td>
<td>76,584</td>
</tr>
</tbody>
</table>

• Baseline Methods: vanilla graph mining algorithm
• Similarity Matrix: Jaccard index, cosine similarity
## Experimental Settings

### Metrics

<table>
<thead>
<tr>
<th>RQ</th>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>$\text{Diff} = \frac{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$KL(\frac{Y^*}{</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\text{Prec@50}$</td>
<td>precision</td>
</tr>
<tr>
<td></td>
<td>$\text{NDCG@50}$</td>
<td>normalized discounted cumulative gain</td>
</tr>
<tr>
<td>spectral clustering</td>
<td>$\text{NMI}(\mathcal{C}_{Y^*}, \mathcal{C}_Y)$</td>
<td>normalized mutual information</td>
</tr>
<tr>
<td></td>
<td>$\text{ROC} - AUC(Y^*, \bar{Y})$</td>
<td>area under ROC curve</td>
</tr>
<tr>
<td></td>
<td>$F1(Y^*, \bar{Y})$</td>
<td>F1 score</td>
</tr>
<tr>
<td>RQ2</td>
<td>$\text{Reduce} = 1 - \frac{\text{Tr}((Y^<em>)'L_SY^</em>)}{\text{Tr}(Y'\bar{Y}L_S\bar{Y})}$</td>
<td>degree of reduce in individual bias</td>
</tr>
<tr>
<td>RQ3</td>
<td>Running time in seconds</td>
<td>running time</td>
</tr>
</tbody>
</table>
Experimental Results

Table 1: Effectiveness results for PageRank. Lower is better in gray columns. Higher is better in the others.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Jaccard Index</th>
<th>Cosine Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diff</td>
<td>KL</td>
</tr>
<tr>
<td>Twitch</td>
<td>0.109</td>
<td>5.37 × 10^{-3}</td>
</tr>
<tr>
<td>PPI</td>
<td>0.185</td>
<td>1.90 × 10^{-3}</td>
</tr>
</tbody>
</table>

• **Obs.**: 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)
Roadmap

• Motivations ✓
• InFoRM Measures ✓
• InFoRM Algorithms ✓
  – Debiasing the Input Graph
  – Debiasing the Mining Model
  – Debiasing the Mining Results
• InFoRM Cost ✓
• Experimental Results ✓
• Conclusions
Conclusions

• **Problem:** InFoRM (individual fairness on graph mining)
  – **fundamental questions:** measures, algorithms, cost

• **Solutions:**
  – **Measures:** $\text{Bias}(Y, S) = \text{Tr}(Y'SY)$
  – **Algorithms:** debiasing (1) the input graph, (2) the mining model and (3) the mining results
  – **Cost:** the upper bound of $\|\bar{Y} - Y^*\|_F$
    • Upper bound on debiasing the mining results
    • Case study on debiasing PageRank algorithm

• **Results:** effective in mitigating individual bias in the graph mining results while maintaining the performance of vanilla algorithm

• More details in the paper
  – proofs and analysis
  – detailed experimental settings
  – additional experimental results

---

Table 2: Effectiveness results for spectral clustering. Lower is better in gray columns. Higher is better in the others.

<table>
<thead>
<tr>
<th>Debiasing the Input Graph</th>
<th>Jaccard Index</th>
<th>Cosine Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datasets</strong></td>
<td>Diff</td>
<td>NMI</td>
</tr>
<tr>
<td>Twitch</td>
<td>0.031</td>
<td>1.000</td>
</tr>
<tr>
<td>PPI</td>
<td>1.035</td>
<td>0.914</td>
</tr>
</tbody>
</table>