InFoRM: Individual Fairness on Graph Mining

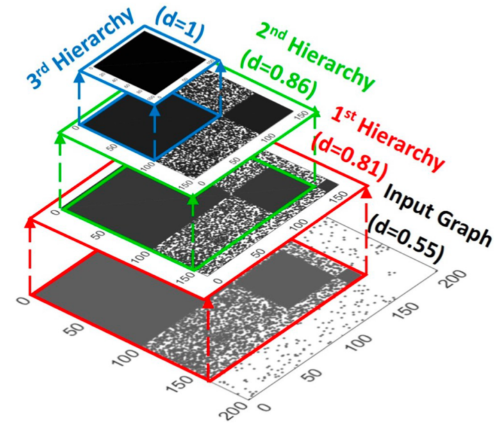
Jian Kang Jingrui He Ross Maciejewski Hanghang Tong




ILLINOIS

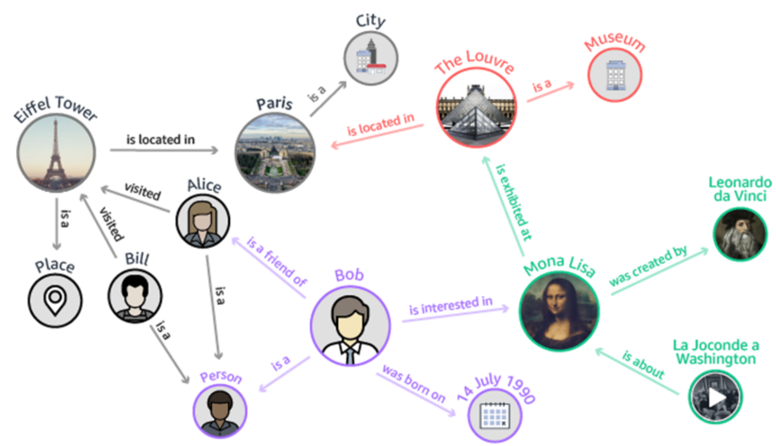
Graph Mining: Applications

Social Science [1]



Finance [2]

Biology [3]



Cognitive Science [4]

[1] Borgatti, S. P., Mehra, A., Brass, D. J., & Labianca, G.. Network Analysis in the Social Sciences. Science 2009.

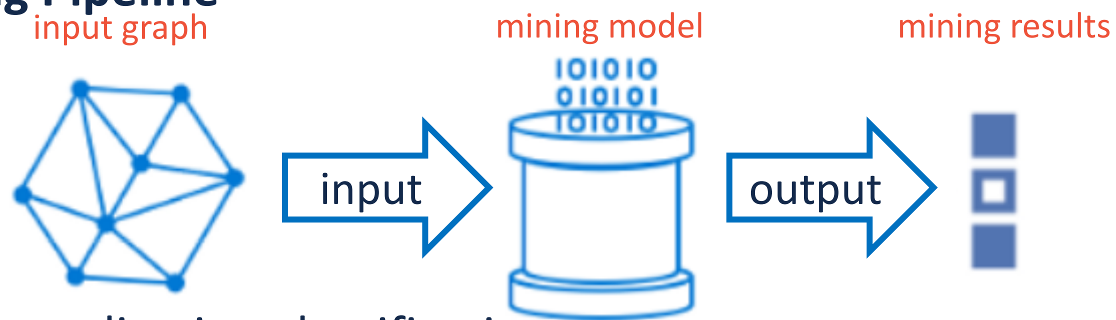
[2] Zhang, S., Zhou, D., Yildirim, M. Y., Alcorn, S., He, J., Davulcu, H., & Tong, H.. Hidden: Hierarchical Dense Subgraph Detection with Application to Financial Fraud Detection. SDM 2017.

[3] Wang, S., He, L., Cao, B., Lu, C. T., Yu, P. S., & Ragin, A. B.. Structural Deep Brain Network Mining. KDD 2017.

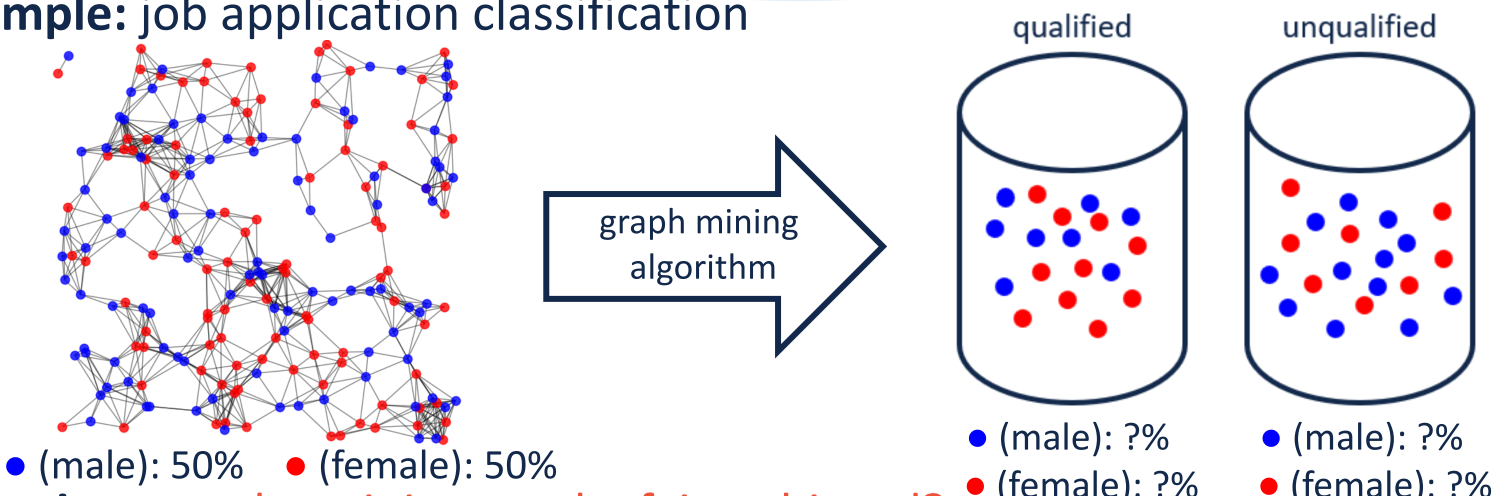
[4] Ding, M., Zhou, C., Chen, Q., Yang, H., & Tang, J.. Cognitive Graph for Multi-Hop Reading Comprehension at Scale. ACL 2019.

Graph Mining: How To

- **Graph Mining Pipeline**



- **Example:** job application classification



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

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]

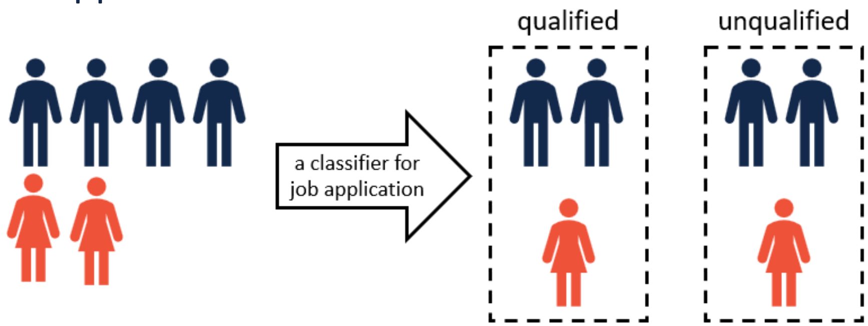
[1] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S.. Certifying and Removing Disparate Impact. KDD 2015.
[2] Chouldechova, A., & Roth, A.. The Frontiers of Fairness in Machine Learning. arXiv.
[3] Hardt, M., Price, E., & Srebro, N.. Equality of Opportunity in Supervised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

Group Fairness: Statistical Parity

- **Definition:** candidates in protected and unprotected groups have equal probability of being assigned to a predicted class c

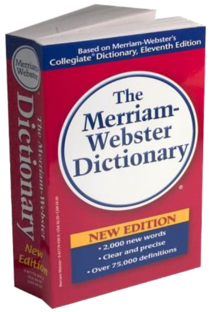
$$\Pr_+(y = c) = \Pr_-(y = c)$$

  - $\Pr_+(y = c)$: probability of being assigned to c for protected group; $\Pr_-(y = c)$ 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

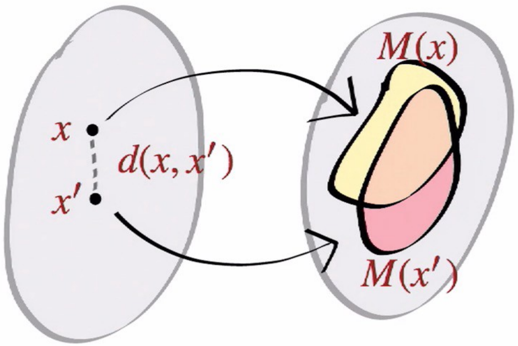
Individual Fairness



- **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 $\mathcal{D}$

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

- **Illustrative Example:**



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

[1] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R.. Fairness through Awareness. ITCS 2012.
[2] https://www.merriam-webster.com/dictionary/fairness

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

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

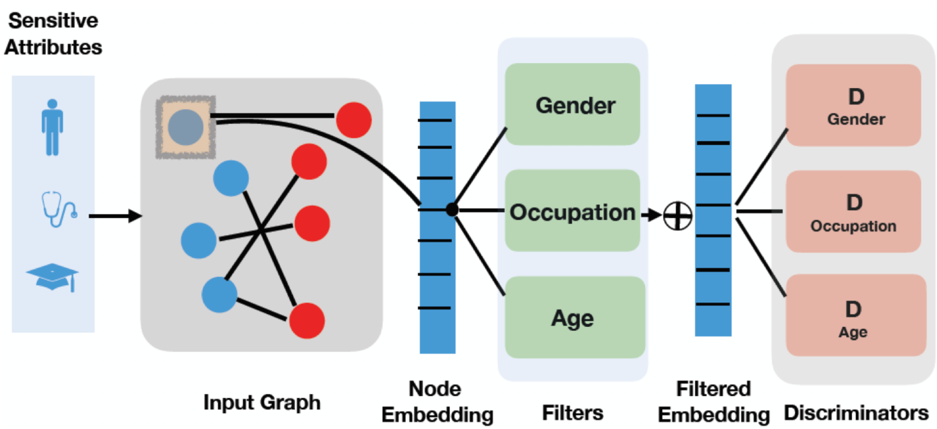
Algorithmic Fairness in Graph Mining

- **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

[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.
[2] Rahman, T. A., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.
[3] Bose, A. J., & Hamilton, W. L.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.
[4] Palowitch, J., & Perozzi, B.. Monet: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit. arXiv.
[5] Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J.. Enhancement of the Neutrality in Recommendation. RecSys 2012 Workshop.
[6] Yao, S., & Huang, B.. Beyond Parity: Fairness Objectives for Collaborative Filtering. NIPS 2017.

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



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

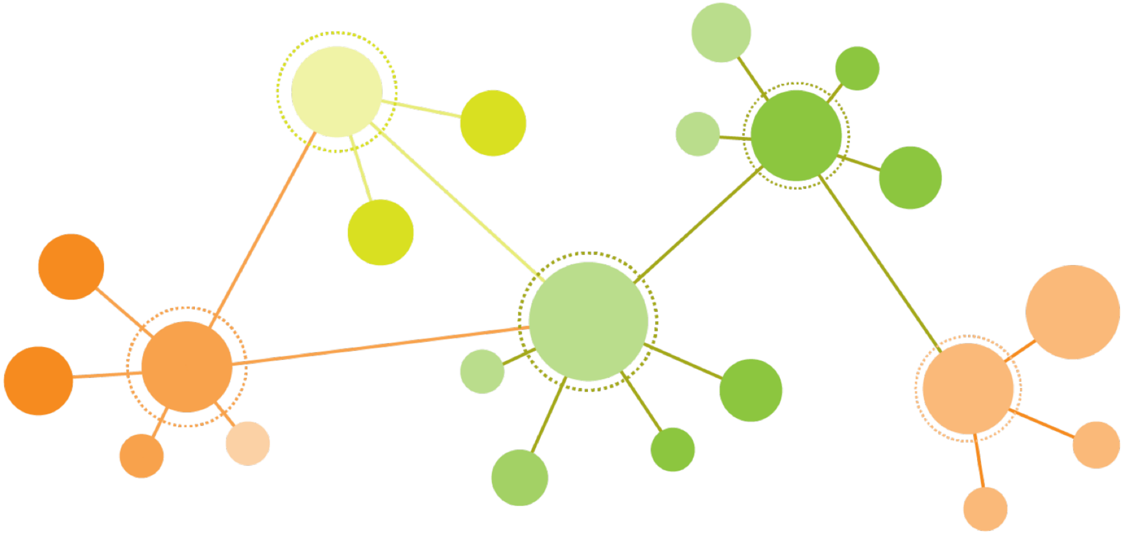
Algorithmic Fairness in Graph Mining

- **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!

[1] Kleindessner, M., Samadi, S., Awasthi, P., & Morgenstern, J.. Guarantees for Spectral Clustering with Fairness Constraints. ICML 2019.
[2] Rahman, T. A., Surma, B., Backes, M., & Zhang, Y.. Fairwalk: Towards Fair Graph Embedding. IJCAI 2019.
[3] Bose, A. J., & Hamilton, W. L.. Compositional Fairness Constraints for Graph Embeddings. ICML 2019.
[4] Palowitch, J., & Perozzi, B.. Monet: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit. arXiv.
[5] Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J.. Enhancement of the Neutrality in Recommendation. RecSys 2012 Workshop.
[6] Yao, S., & Huang, B.. Beyond Parity: Fairness Objectives for Collaborative Filtering. NIPS 2017.

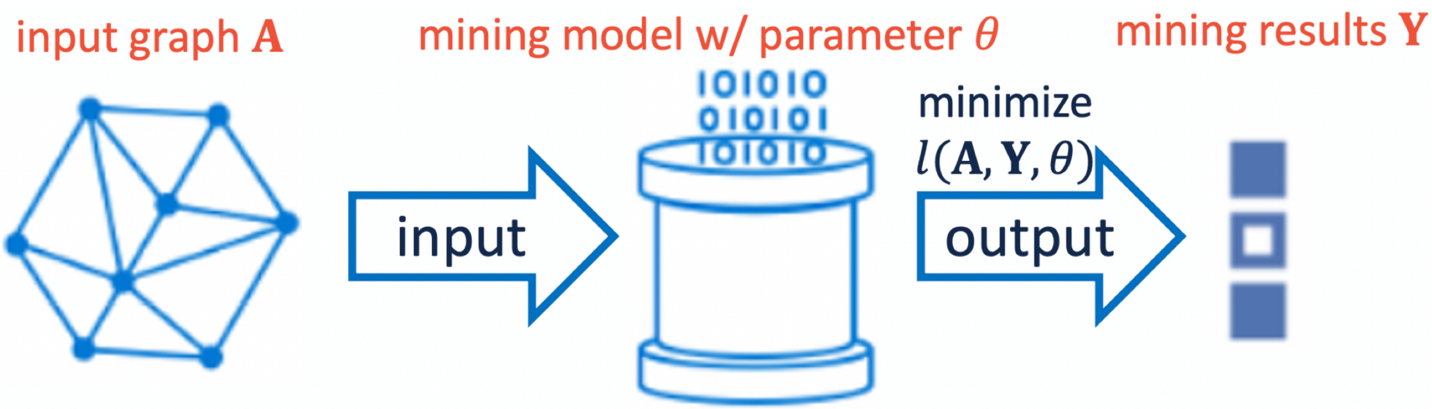
InFoRM: Individual Fairness on Graph Mining

- **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**



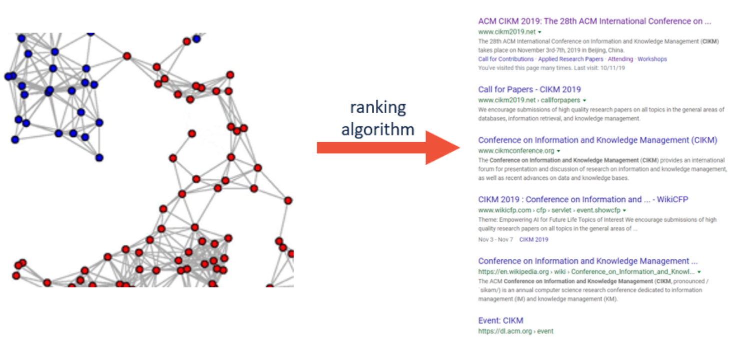
  - **Input:**
    - Input graph $\mathbf{A}$
    - Model parameters θ
  - **Output:** mining results $\mathbf{Y}$
    - Examples: ranking vectors, class probabilities, embeddings

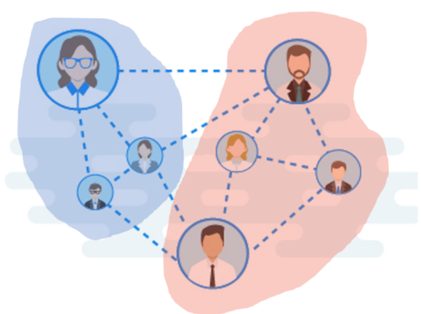
minimize loss function
$l(\mathbf{A}, \mathbf{Y}, \theta)$

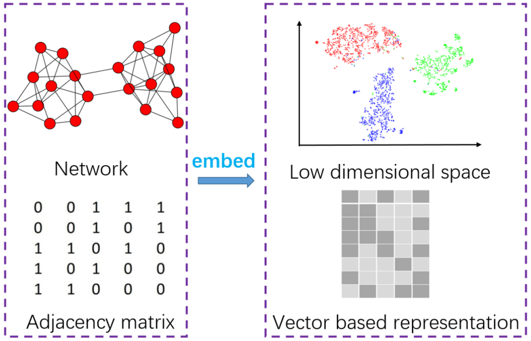
Classic Graph Mining Algorithms

Examples of Classic Graph Mining Algorithm

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





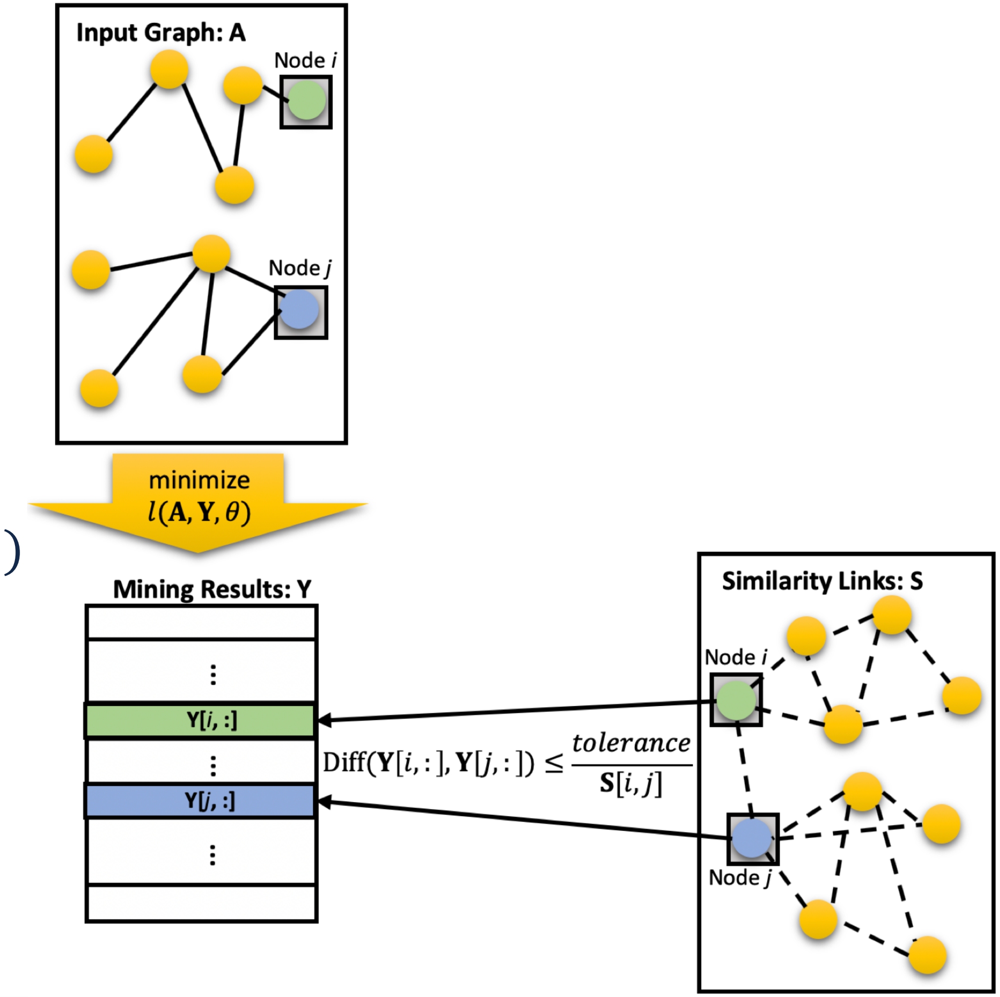


Roadmap

- 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 $\mathbf{S}$
    - 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 results are fair
  - individual bias measure $\mathrm{Bias}(\mathbf{Y}, \mathbf{S})$

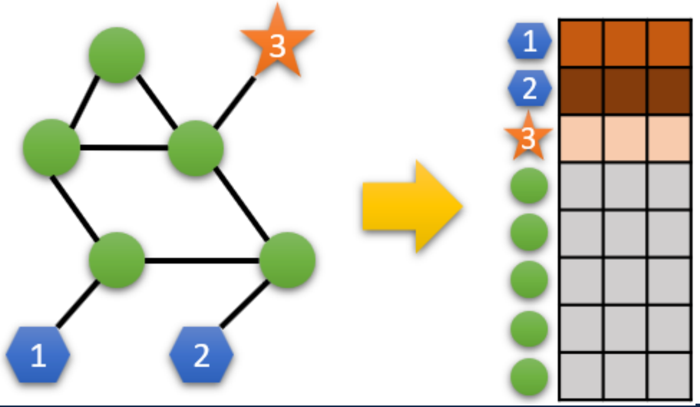


Measuring Individual Bias: Formulation

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

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

  - **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
    - **Problem:** too restrictive to be fulfilled
- **Relaxed Criteria:** $\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{Y}[i,:]-\mathbf{Y}[j,:]\|_F^2\mathbf{S}[i,j] = 2\mathrm{Tr}(\mathbf{Y}'\mathbf{L_S}\mathbf{Y}) \le m\epsilon = \delta$



Measuring Individual Bias: Solution

- **InFoRM (Individual Fairness on Graph Mining)**
  - Given (1) a graph mining results $\mathbf{Y}$, (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

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

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

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

Lipschitz Property of Individual Fairness

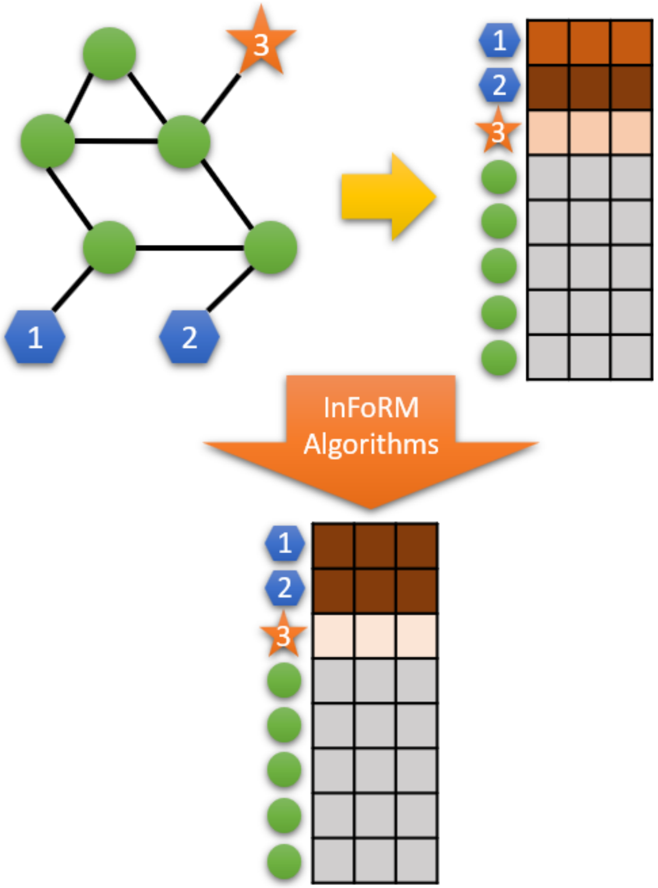
- **Connection to Lipschitz Property**
  - $(\boldsymbol{D_1}, \boldsymbol{D_2})$**-Lipschitz property [1]:** a function f is (D_1, D_2)-Lipschitz if it satisfies
    $$D_1\big(f(i), f(j)\big) \leq L D_2(i, j), \forall (x, y)$$
    - L is Lipschitz constant
  - InFoRM naturally satisfies (D_1, D_2)-Lipschitz property as long as
    - $f(i) = \mathbf{Y}[i, :]$
    - $D_1\big(f(i), f(j)\big) = \|\mathbf{Y}[i, :] - \mathbf{Y}[j, :]\|_2^2$, $D_2(i, j) = \frac{1}{\mathbf{S}[i,j]}$
  - Lipschitz constant of InFoRM is ϵ

Roadmap

- Motivations
- InFoRM Measures
- InFoRM Algorithms
  - Debiasing the Input Graph
  - Debiasing the Mining Model
  - Debiasing the Mining Results
- InFoRM Cost
- Experimental Results
- Conclusions

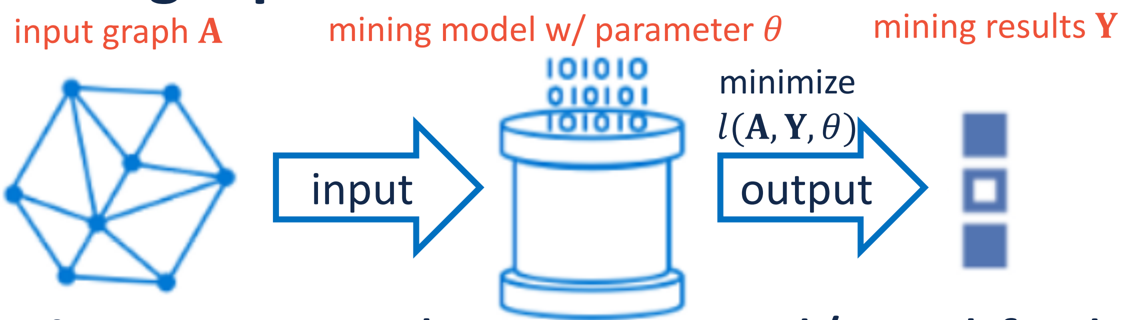
Problem Definition: InFoRM Algorithms

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



Mitigating Individual Bias: How To

- **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

Debiasing the Input Graph

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

consistency in graph topology

$$\min_{\mathbf{Y}} \quad J = \left\|\widetilde{\mathbf{A}} - \mathbf{A}\right\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}'\mathbf{L}_{\mathbf{S}}\mathbf{Y})$$

$$\text{s.t.} \quad \mathrm{Y} = \mathrm{argmin}_{\mathbf{Y}} \; l(\widetilde{\mathbf{A}}, \mathbf{Y}, \theta)$$

bias measure

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

[1] Kang, J., & Tong, H.. N2N: Network Derivative Mining. CIKM 2019.
[2] Mei, S., & Zhu, X.. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. AAAI 2015.

Debiasing the Input Graph

- Considering the KKT conditions,

$$\min_{\mathbf{Y}} \quad J = \left\|\widetilde{\mathbf{A}} - \mathbf{A}\right\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}'\mathbf{L}_{\mathbf{S}}\mathbf{Y})$$

$$\text{s.t.} \quad \partial_{\mathbf{Y}} l\left(\widetilde{\mathbf{A}}, \mathbf{Y}, \theta\right) = 0$$

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

Debiasing the Input Graph

- **Calculating Gradient**

key component to calculate

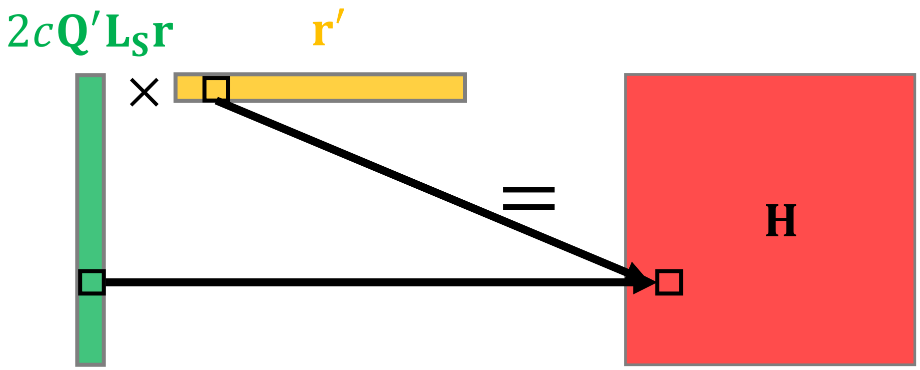
$$\frac{\partial J}{\partial \mathbf{A}} = 2(\tilde{\mathbf{A}} - \mathbf{A}) + \alpha \boxed{\left[\mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L}_{\mathbf{S}} \frac{\partial \tilde{\mathbf{Y}}}{\partial \mathbf{A}[i,j]}\right)\right]}$$

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

  - $\tilde{\mathbf{Y}}$ satisfies $\partial_{\mathbf{Y}} l\left(\tilde{\mathbf{A}}, \mathbf{Y}, \theta\right) = 0$
  - $\mathbf{H} = \left[\mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L}_{\mathbf{S}} \frac{\partial \tilde{\mathbf{Y}}}{\partial \mathbf{A}[i,j]}\right)\right]$ is a matrix with $\mathbf{H}[i,j] = \mathrm{Tr}\left(2\tilde{\mathbf{Y}}\mathbf{L}_{\mathbf{S}} \frac{\partial \tilde{\mathbf{Y}}}{\partial \mathbf{A}[i,j]}\right)$
- **Question:** how to efficiently calculate $\mathbf{H}$?

Instantiation #1: PageRank

- **Goal:** efficiently calculate $\mathbf{H}$ for PageRank
- **Mining Results Y:** $\mathbf{r} = (1 - c)\mathbf{Q}\mathbf{e}$
- **Partial Derivatives H:** $\mathbf{H} = 2c\mathbf{Q}'\mathbf{L}_{\mathbf{S}}\mathbf{r}\mathbf{r}'$
- **Remarks:** $\mathbf{Q} = (\mathbf{I} - c\mathbf{A})^{-1}$
- **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

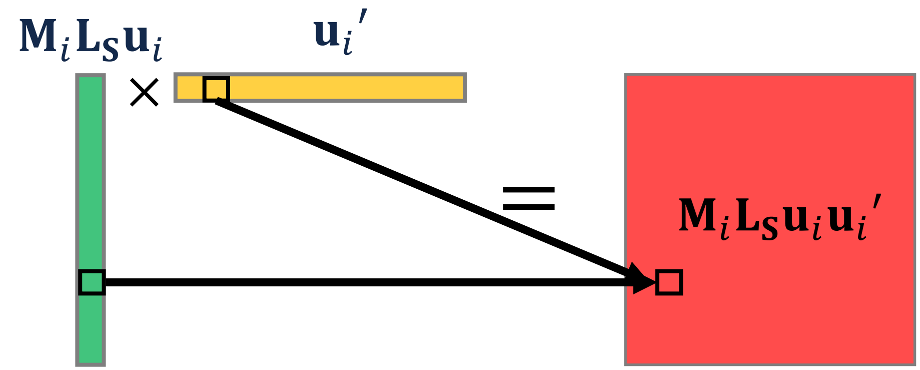


Instantiation #2: Spectral Clustering

- **Goal:** efficiently calculate $\mathbf{H}$ for spectral clustering
- **Mining Results $\mathbf{Y}$:** $\mathbf{U}$ = eigenvectors with k smallest eigenvalues
- **Partial Derivatives $\mathbf{H}$:** $\mathbf{H} = 2\sum_{i=1}^{k}\left(\boxed{\mathrm{diag}(\mathbf{M}_i\mathbf{L}_\mathbf{S}\mathbf{u}_i\mathbf{u}_i{}')\mathbf{1}_{n\times n}} - \boxed{\mathbf{M}_i\mathbf{L}_\mathbf{S}\mathbf{u}_i\mathbf{u}_i{}'}\right)$
- **Remarks:** $(\lambda_i, \mathbf{u}_i) = i$-th smallest eigenpair, $\mathbf{M}_i = (\lambda_i\mathbf{I} - \mathbf{L}_\mathbf{A})^{+}$
- **Time Complexity**
  - Straightforward: $O(k^2(m+n) + k^3n + kn^3)$
  - Ours: $O(k^2(m+n) + k^3n)$

low-rank

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



Instantiation #3: LINE (1st)

- **Goal:** efficiently calculate $\mathbf{H}$ for LINE (1st)
- **Mining Results $\mathbf{Y}$:** $\mathbf{Y}[i,:]\mathbf{Y}[j,:]' = \log \frac{T(\widetilde{\mathbf{A}}[i,j]+\widetilde{\mathbf{A}}[j,i])}{d_i d_j^{3/4}+d_i^{3/4} d_j} - \log b$
  - $d_i =$ outdegree of node i, $T = \sum_{i=1}^{n} d_i^{3/4}$ and $b =$ number of negative samples
- **Partial Derivatives $\mathbf{H}$: $\mathbf{H} =$** $\boxed{2f\left(\widetilde{\mathbf{A}} + \widetilde{\mathbf{A}}'\right) \circ \mathbf{L_S}} - \boxed{2\mathrm{diag}(\mathbf{B}\mathbf{L_S})\mathbf{1}_{n\times n}}$

element-wise in-place calculation

vectorize $\mathrm{diag}(\mathbf{B}\mathbf{L_S})$ and stack it n times

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

stack $\mathbf{d}$ n times

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

task-specific loss function

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

bias measure

- **Solution**
  - **General:** solve by (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:** 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:** $\min_{\mathbf{r}} c\mathbf{r}'(\mathbf{I}-\mathbf{A})\mathbf{r} + (1-c)\|\mathbf{r}-\mathbf{e}\|_F^2 + \alpha\mathbf{r}'\mathbf{L_S}\mathbf{r}$
  - **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:** $\min_{\mathbf{U}} \mathrm{Tr}(\mathbf{U}'\mathbf{L_A}\mathbf{U}) + \alpha\mathrm{Tr}(\mathbf{U}'\mathbf{L_S}\mathbf{U}) = \mathrm{Tr}(\mathbf{U}'\mathbf{L}_{\mathbf{A}+\alpha\mathbf{S}}\mathbf{U})$
  - **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:** $\max_{\mathbf{x}_i,\mathbf{x}_j} \log g(\mathbf{x}_j\mathbf{x}_i') + b\mathbb{E}_{j'\in P_n}\left[\log g\left(-\mathbf{x}_{j'}\mathbf{x}_i'\right)\right] - \alpha\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_F^2\mathbf{S}[i,j]$
    $\forall i,j = 1,\dots,n$
  - **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**

consistency of mining results, convex

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

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 $\alpha \geq 0 \rightarrow$ 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

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 $\bar{\mathbf{Y}}$
  - Debiased mining results $\mathbf{Y}^*$
    - 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

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 $\mathbf{A}$
  - A node-node similarity matrix $\mathbf{S}$
  - Vanilla mining results $\bar{\mathbf{Y}}$
  - Debiased mining results $\mathbf{Y}^* = (\mathbf{I} + \alpha\mathbf{S})^{-1}\bar{\mathbf{Y}}$
- If $||\mathbf{S} - \mathbf{A}||_F = \delta$, we have

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

- **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 $\mathbf{A}$ ⟶ could be small due to low-rank structures in real-world graphs
  - The largest singular value of $\mathbf{A}$ ⟶ could be small if $\mathbf{A}$ 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 $\mathbf{r}^* = (\mathbf{I} + \alpha\mathbf{S})^{-1}\bar{\mathbf{Y}}$
- If $\|\mathbf{S} - \mathbf{A}\|_F = \delta$, we have

$$\|\bar{\mathbf{r}} - \mathbf{r}^*\|_F \leq \frac{2\alpha n}{1-c}\left(\delta + \sqrt{rank(\mathbf{A})}\sigma_{\max}(\mathbf{A})\right)$$

- **Observation:** the cost of debiasing PageRank depends on
  - The number of nodes n (i.e. size of the input graph)
  - The difference δ between $\mathbf{A}$ and $\mathbf{S}$
  - The rank of $\mathbf{A}$ $\longrightarrow$ could be small due to low-rank structures in real-world graphs
  - The largest singular value of $\mathbf{A}$ $\longrightarrow$ 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

Name	Nodes	Edges
AstroPh	18,772	198,110
CondMat	23,133	93,497
Facebook	22,470	171,002
Twitter	7,126	35,324
PPI	3,890	76,584

- **Baseline Methods:** vanilla graph mining algorithm
- **Similarity Matrix:** Jaccard index, cosine similarity

Experimental Settings

- Metrics

<table>
<tr><th></th><th colspan="2">Metric</th><th>Definition</th></tr>
<tr><td rowspan="7">RQ1</td><td colspan="2">$\mathrm{Diff} = \frac{\|\mathbf{Y}^{*} - \bar{\mathbf{Y}}\|_F}{\|\bar{\mathbf{Y}}\|_F}$</td><td>difference between fair and vanilla graph mining results</td></tr>
<tr><td rowspan="3">PageRank</td><td>$KL\left(\frac{\mathbf{Y}^{*}}{\|\mathbf{Y}^{*}\|_1} \,\|\, \frac{\bar{\mathbf{Y}}}{\|\bar{\mathbf{Y}}\|_1}\right)$</td><td>KL divergence</td></tr>
<tr><td>$Prec@50$</td><td>precision</td></tr>
<tr><td>$NDCG@50$</td><td>normalized discounted cumulative gain</td></tr>
<tr><td>spectral clustering</td><td>$NMI(\mathcal{C}_{\mathbf{Y}^{*}}, \mathcal{C}_{\bar{\mathbf{Y}}})$</td><td>normalized mutual information</td></tr>
<tr><td rowspan="2">LINE</td><td>$ROC-AUC(\mathbf{Y}^{*}, \bar{\mathbf{Y}})$</td><td>area under ROC curve</td></tr>
<tr><td>$F1(\mathbf{Y}^{*}, \bar{\mathbf{Y}})$</td><td>F1 score</td></tr>
<tr><td>RQ2</td><td colspan="2">$Reduce = 1 - \frac{\mathrm{Tr}\left((\mathbf{Y}^{*})' \, \mathbf{L_S} \mathbf{Y}^{*}\right)}{\mathrm{Tr}(\bar{\mathbf{Y}}' \mathbf{L_S} \bar{\mathbf{Y}})}$</td><td>degree of reduce in individual bias</td></tr>
<tr><td>RQ3</td><td colspan="2">Running time in seconds</td><td>running time</td></tr>
</table>

Experimental Results

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

Debiasing the Input Graph

Datasets	Jaccard Index						Cosine Similarity					
	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

Datasets	Jaccard Index						Cosine Similarity					
	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

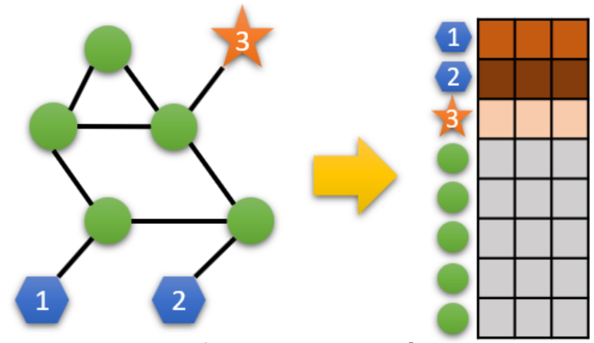
Datasets	Jaccard Index						Cosine Similarity					
	Diff	KL	Prec@50	NDCG@50	Reduce	Time	Diff	KL	Prec@50	NDCG@50	Reduce	Time
Twitch	0.035	9.75×10^{-4}	0.980	0.986	33.9%	0.033	0.101	5.84×10^{-3}	0.940	0.958	44.6%	0.024
PPI	0.045	1.22×10^{-3}	0.940	0.958	27.0%	0.020	0.112	6.97×10^{-3}	0.940	0.958	45.0%	0.019

- **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)

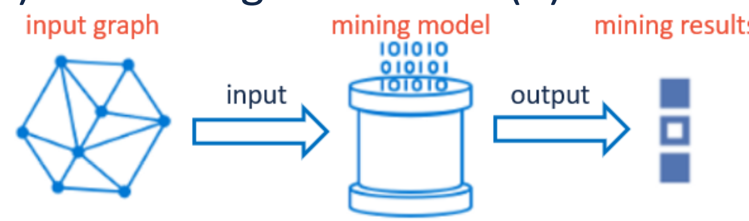
Roadmap

- 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:** $\mathrm{Bias}(\mathbf{Y}, \mathbf{S}) = \mathrm{Tr}(\mathbf{Y}'\mathbf{S}\mathbf{Y})$
  - **Algorithms:** debiasing (1) the input graph, (2) the mining model and (3) the mining results
  - **Cost:** the upper bound of $\|\overline{\mathbf{Y}} - \mathbf{Y}^*\|_F$
    - 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.

Debiasing the Input Graph								
Datasets	Jaccard Index				Cosine Similarity			
	Diff	NMI	Reduce	Time	Diff	NMI	Reduce	Time
Twitch	0.031	1.000	5.44%	1698	0.107	1.000	24.5%	1714
PPI	1.035	0.914	19.5%	829.3	0.933	0.849	24.1%	985.1