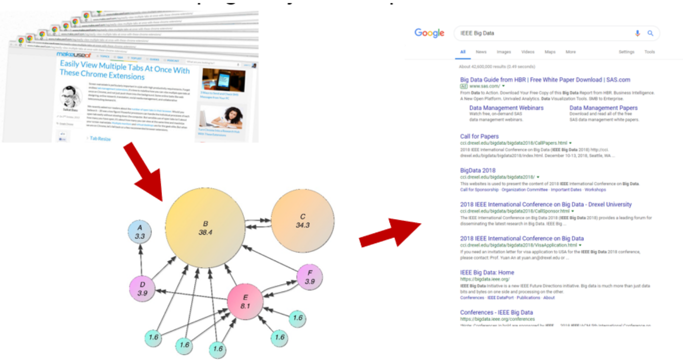
N2N: Network Derivative Mining

Jian Kang Hanghang Tong

ILLINOIS

Network Mining: Applications

- Network mining is ubiquitous.



Information Retrieval [Weng 2010]

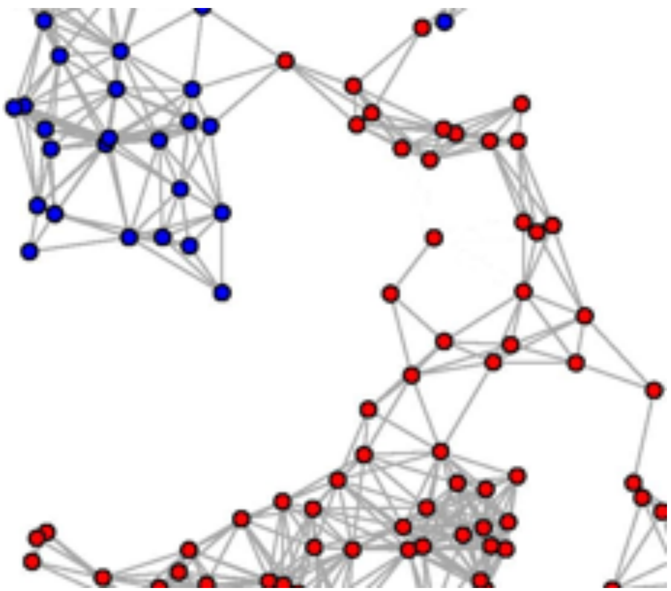
Recommender System [Gori et al. 2007]

Social Network Analysis [Tang et al. 2010]

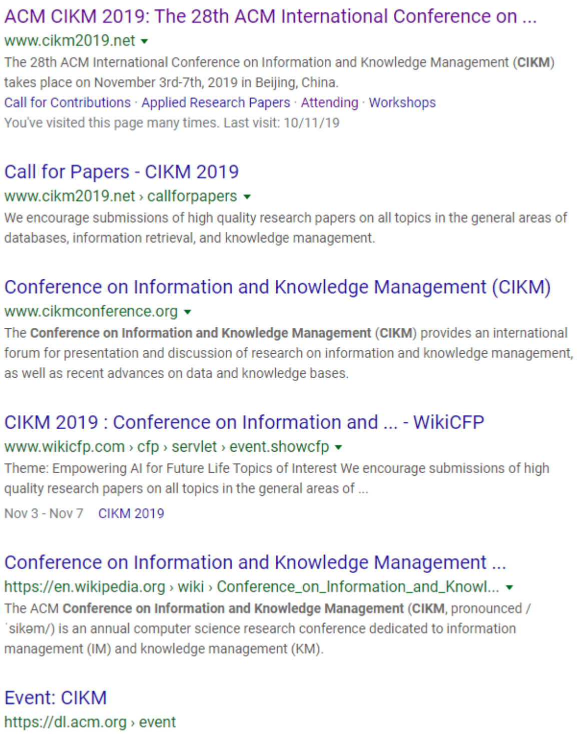
[1] Weng, J., Lim, E.-P., Jiang, J. & He, Q.. Twitterrank: Finding Topic-Sensitive Influential Twitterers. WSDM 2010.
[2] Gori, M., & Pucci, A. ItemRank: A Random-walk Based Scoring Algorithm for Recommendation Engines. IJCAI 2007.
[3] Tang, L., & Liu, H. Graph Mining Applications to Social Network Analysis. Managing and Mining Graph Data 2010.

Network Mining: Ranking

- **Information Retrieval:** which webpages are the most important in WWW?



ranking algorithm



ACM CIKM 2019: The 28th ACM International Conference on ...
www.cikm2019.net ▾
The 28th ACM International Conference on Information and Knowledge Management (**CIKM**) takes place on November 3rd-7th, 2019 in Beijing, China.
Call for Contributions · Applied Research Papers · Attending · Workshops
You've visited this page many times. Last visit: 10/11/19

Call for Papers - CIKM 2019
www.cikm2019.net › callforpapers ▾
We encourage submissions of high quality research papers on all topics in the general areas of databases, information retrieval, and knowledge management.

Conference on Information and Knowledge Management (CIKM)
www.cikmconference.org ▾
The **Conference on Information and Knowledge Management** (**CIKM**) provides an international forum for presentation and discussion of research on information and knowledge management, as well as recent advances on data and knowledge bases.

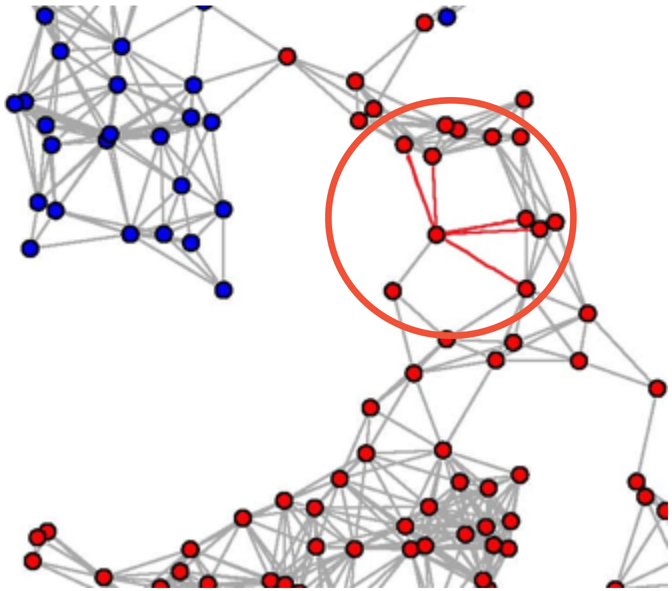
CIKM 2019 : Conference on Information and ... - WikiCFP
www.wikicfp.com › cfp › servlet › event.showcfp ▾
Theme: Empowering AI for Future Life Topics of Interest We encourage submissions of high quality research papers on all topics in the general areas of ...
Nov 3 - Nov 7 CIKM 2019

Conference on Information and Knowledge Management ...
https://en.wikipedia.org › wiki › Conference_on_Information_and_Knowl... ▾
The ACM **Conference on Information and Knowledge Management** (**CIKM**, pronounced /ˈsikəm/) is an annual computer science research conference dedicated to information management (IM) and knowledge management (KM).

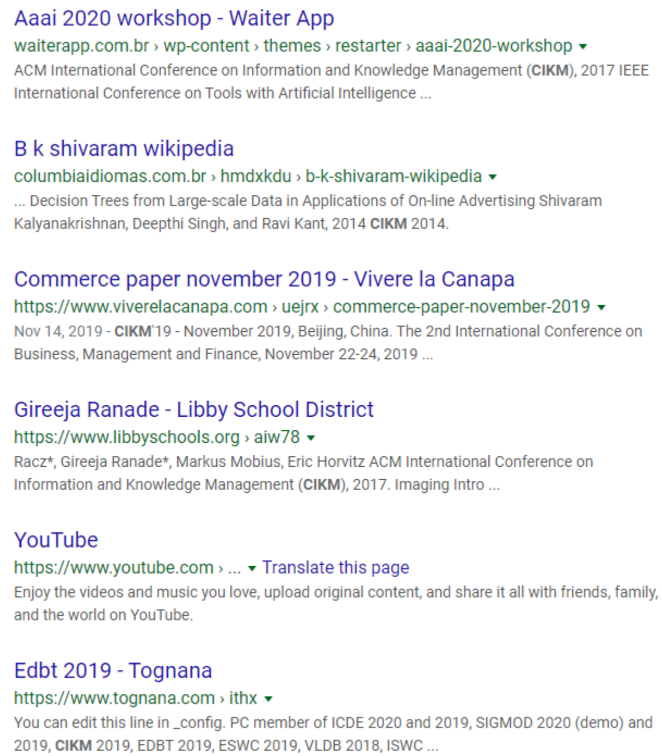
Event: CIKM
https://dl.acm.org › event

Network Mining: Ranking

- **Question:** **WHY** does ranking algorithm act sensitive to malicious links?



ranking algorithm



Aaai 2020 workshop - Waiter App
waiterapp.com.br › wp-content › themes › restarter › aaai-2020-workshop
ACM International Conference on Information and Knowledge Management (**CIKM**), 2017 IEEE International Conference on Tools with Artificial Intelligence ...

B k shivaram wikipedia
columbiaidiomas.com.br › hmdxkdu › b-k-shivaram-wikipedia
... Decision Trees from Large-scale Data in Applications of On-line Advertising Shivaram Kalyanakrishnan, Deepthi Singh, and Ravi Kant, 2014 **CIKM** 2014.

Commerce paper november 2019 - Vivere la Canapa
https://www.viverelacanapa.com › uejrx › commerce-paper-november-2019
Nov 14, 2019 - **CIKM**'19 - November 2019, Beijing, China. The 2nd International Conference on Business, Management and Finance, November 22-24, 2019 ...

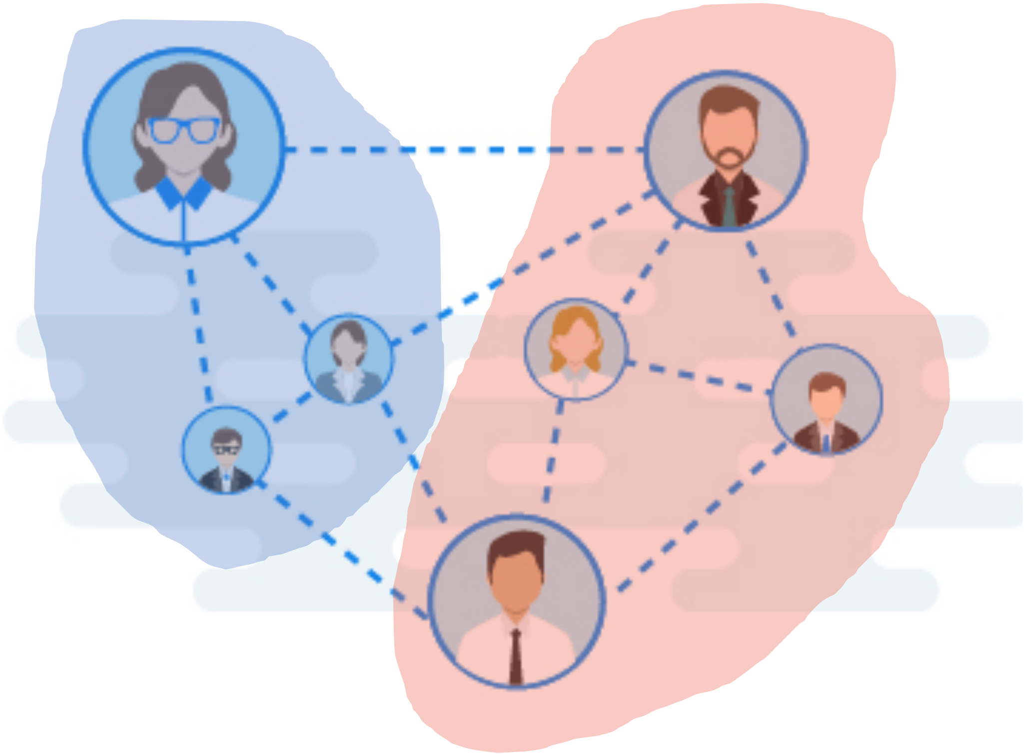
Gireeja Ranade - Libby School District
https://www.libbyschools.org › aiw78
Racz*, Gireeja Ranade*, Markus Mobius, Eric Horvitz ACM International Conference on Information and Knowledge Management (**CIKM**), 2017. Imaging Intro ...

YouTube
https://www.youtube.com › ... Translate this page
Enjoy the videos and music you love, upload original content, and share it all with friends, family, and the world on YouTube.

Edbt 2019 - Tognana
https://www.tognana.com › ithx
You can edit this line in _config. PC member of ICDE 2020 and 2019, SIGMOD 2020 (demo) and 2019, **CIKM** 2019, EDBT 2019, ESWC 2019, VLDB 2018, ISWC ...

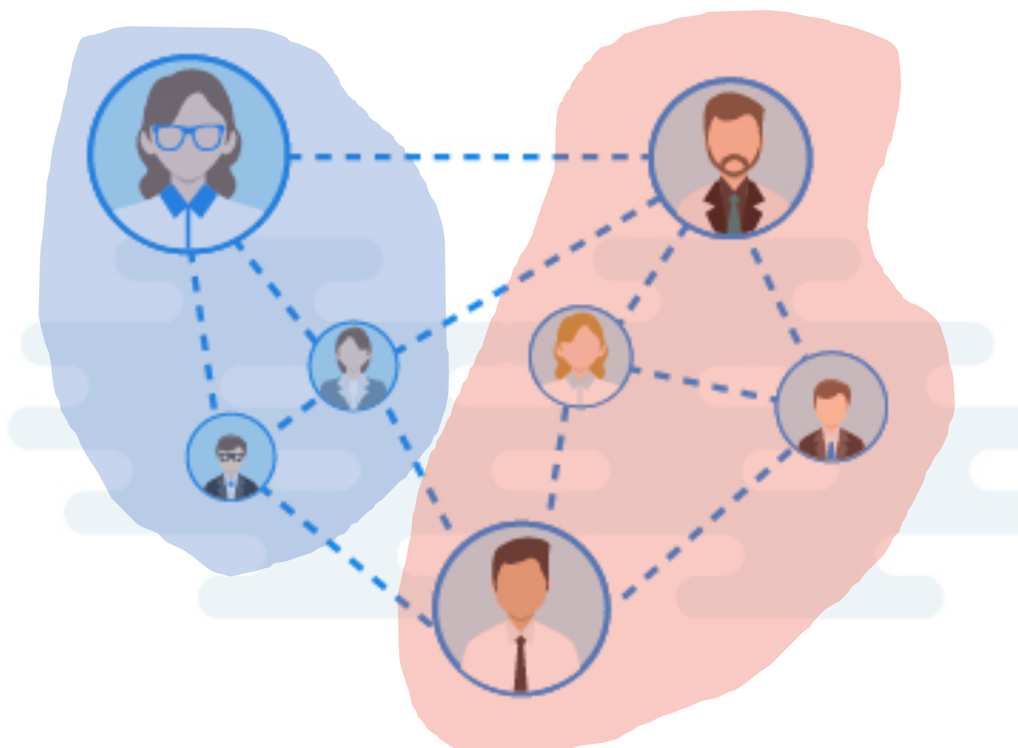
Network Mining: Clustering

- **Clustering:** who will be grouped into the same community?



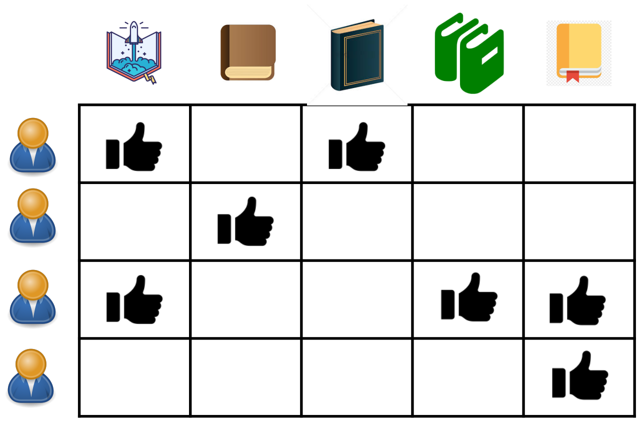
Network Mining: Clustering

- **Question:** **WHY** do  and  belong to the same community?

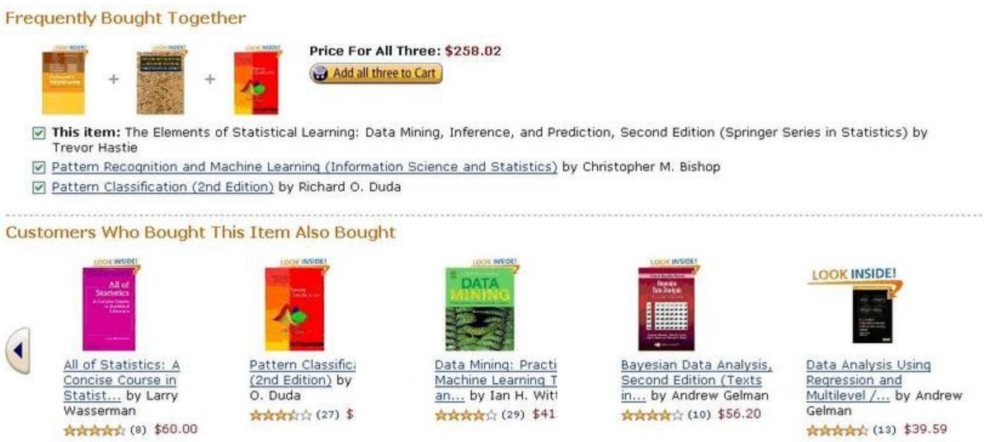


Network Mining: Recommendation

- **Recommendation:** which item best suits a user's taste?

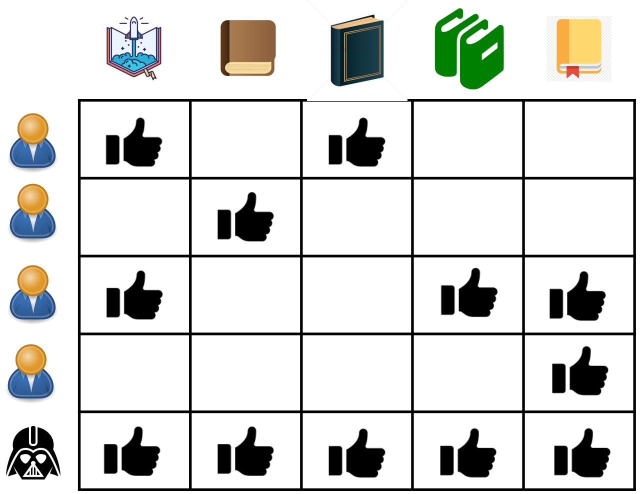


recommender system



Network Mining: Recommendation

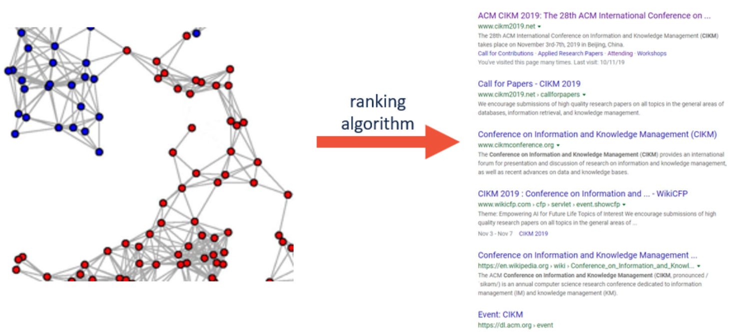
- **Question:** **HOW** do fake ratings affect recommender system?

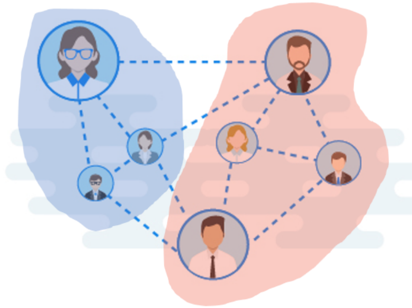


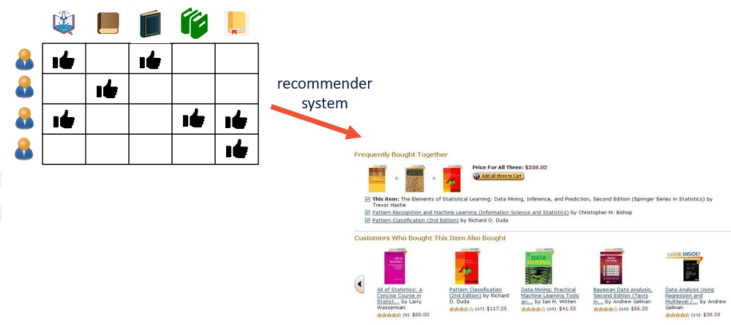
recommender system

Network Mining: Limitations

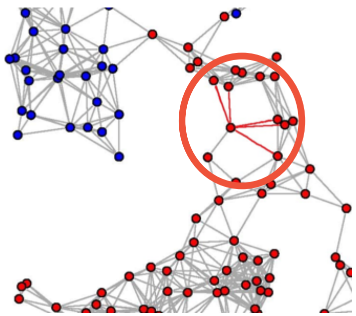
- **Advantage:** good at answer ***what/which*** questions.

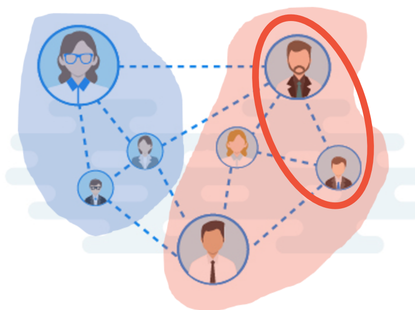


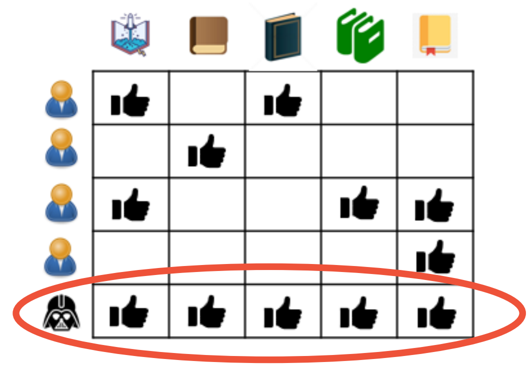




- **Disadvantage:** cannot answer ***why/how*** questions.





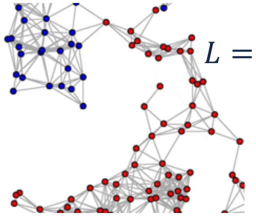


- **Key Question:** how do network mining results relate to underlying structure?

Problem Definition: Network Derivative Mining (N2N)

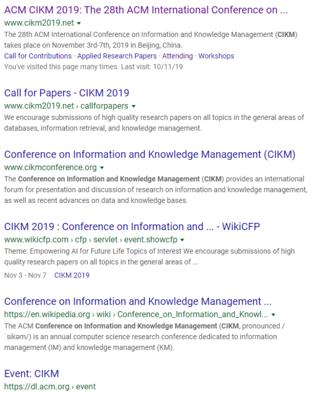
- **N2N:** *network* to *derivative network*
- **Input:**
  - (1) an adjacency matrix $\mathbf{A}$
  - (2) a network mining algorithm $L(\mathbf{A}, Y, \theta)$
    - loss function $L(\cdot)$
    - optimal model output $Y^* = \text{argmin}_Y L(\mathbf{A}, Y, \theta)$
    - additional set of parameters θ
  - (3) a scalar function over optimal model output $f(Y^*)$

input graph $\mathbf{A}$



$L = \|\mathbf{A} - \mathbf{u}\mathbf{v}'\|_F^2$

ranked webpages Y^*



- **Output:** a derivative network $\mathbf{B}$
  - $\mathbf{B}(i, j) =$ influence of edge $\mathbf{A}(i, j)$ on Y^*
  - $\mathbf{B}(i, j) = 0$ if $\mathbf{A}(i, j)$ does not exist

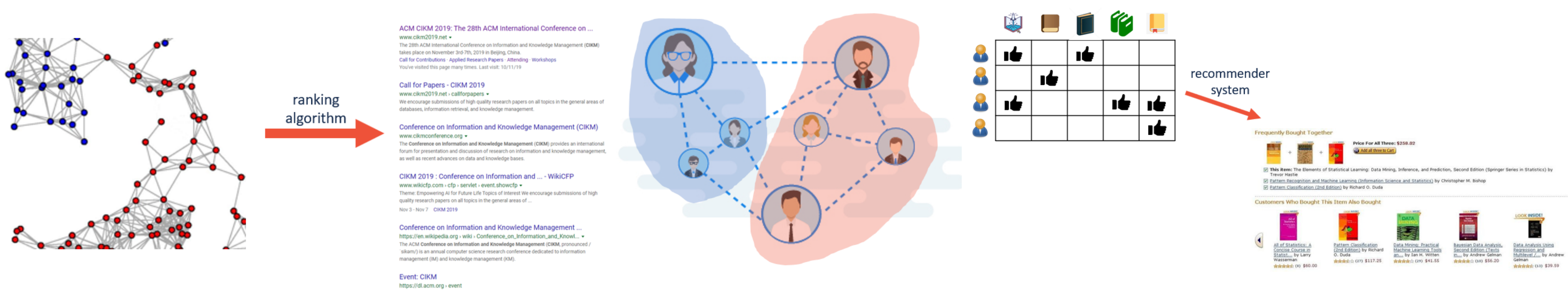
* We focus on the derivatives for the existing edges here, but it naturally applies to non-existing edges.

Problem Definition

Examples of Network Mining Algorithm

Mining Task	Loss Function $L()$	Mining Result Y^*	Parameters	Scalar function $f()$
HITS	$\min_{\mathbf{u},\mathbf{v}} \lVert\mathbf{A}-\mathbf{u}\mathbf{v}'\rVert_F^2$	hubs $\mathbf{u}$ authorities $\mathbf{v}$	none	$f(Y^*)=\lambda_1-\lambda_2$
Spectral Clustering	$\min_{\mathbf{U}} \mathrm{Tr}\,(\mathbf{U}'\mathbf{L}\mathbf{U})$ $\text{s.t.}\quad \mathbf{U}'\mathbf{U}=\mathbf{I}$	eigenvectors $\mathbf{U}$	# clusters k	$f(Y^*)=\mathrm{Tr}(\mathbf{U}'\mathbf{L}\mathbf{U})$
Matrix Completion	$\min_{\mathbf{U},\mathbf{V}} \lVert\mathrm{proj}_\Omega(\mathbf{A}-\mathbf{U}\mathbf{V}')\rVert_F^2 + \lambda_u\lVert\mathbf{U}\rVert_F^2+\lambda_v\lVert\mathbf{V}\rVert_F^2$	user matrix $\mathbf{U}$ item matrix $\mathbf{V}$	latent dimension k λ_u,λ_v	$f(Y^*)=\lVert\mathbf{U}\mathbf{V}'\rVert_F^2$

* The choice of scalar function $f()$ is flexible.



N2N: Formulation

- **Intuition:** influential → high impact if perturbed
- **Definition of Edge Influence:** the derivative of $f(Y^*)$ w.r.t. the edge.

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

- **Mathematical Formulation:**

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

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

key component to calculate

Potential Benefits

- Derivative is a powerful tool.
- **Application Scenarios:**
  - Explainable network mining
  - Adversarial attacks on network mining
  - Active learning
  - Optimal network learning
  - Counterfactual learning

derivative analysis

Challenges

- **C1: Efficiency**
  - Network mining $Y^* = \mathrm{argmin}_Y L(\mathbf{A}, Y, \theta)$
  - Influence calculation $\mathbf{B}(i,j) = \frac{\mathrm{d}f(Y^*)}{\mathrm{d}\mathbf{A}(i,j)}$

  potentially complicated

  - **Question:** how to construct the derivative network efficiently?
- **C2: Scalability**
  - Iterating the influence calculation over all edges.

  possibly super-linear time and space complexities

  - **Question:** how to scale up the derivative network generation ?

Roadmap

- Motivations ✓
- N2N Instantiations
  - Task #1: HITS
  - Task #2: Spectral Clustering
  - Task #3: Matrix Completion
- Experimental Results
- Conclusions

N2N: Algorithmic Framework

- **Input:** the adjacency matrix $\mathbf{A}$, a network mining algorithm $L(\mathbf{A}, Y, \theta)$, a scalar function $f(\cdot)$.
- **Output:** the derivative network $\mathbf{B}$.

3-step strategy:

1. Run network mining algorithm $Y^* = \mathrm{argmin}_Y L(\mathbf{A}, Y, \theta)$
2. Calculate partial derivative $\frac{\partial f(Y^*)}{\partial \mathbf{A}}$
3. Construct derivative network

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

Task #1: HITS

- **Goal:** importance of nodes = hub scores $\mathbf{u}$ + authority scores $\mathbf{v}$

$$\mathbf{u} = \mathbf{A}\mathbf{v}$$
$$\mathbf{v} = \mathbf{A}'\mathbf{u}$$

- **Optimization Problem:**

$$\min_{\mathbf{u},\mathbf{v}} \|\mathbf{A} - \mathbf{u}\mathbf{v}'\|_F^2$$

- **Solution:** rank-1 SVD
  - $\mathbf{u}$ = first left singular vector of $\mathbf{A}$ = principal eigenvector of $\mathbf{A}\mathbf{A}'$
  - $\mathbf{v}$ = first right singular vector of $\mathbf{A}$ = principal eigenvector of $\mathbf{A}'\mathbf{A}$
- **Question:** how does the network topology influence the quality of ranking by HITS?

N2N for HITS

- **Choice of $f(\cdot)$ Function:** $f(Y^*) = \lambda_1 - \lambda_2$
  - Sensitive to eigengap [Ng et al. 2001].
- **Constructing the Derivative Network**:
  - **Chain Rule:**

$$\frac{\partial f(Y^*)}{\partial \mathrm{A}(i,j)} = \mathrm{Tr}\left[\left(\frac{\partial(\lambda_1 - \lambda_2)}{\partial \mathbf{AA}'}\right)' \frac{\partial \mathbf{AA}'}{\partial \mathrm{A}(i,j)}\right] = 2\mathbf{u_1}(i)(\mathbf{u}_1'\mathbf{A})(j) - 2\mathbf{u_2}(i)(\mathbf{u}_2'\mathbf{A})(j)$$

  - **Matrix Form Solution:**

$$\frac{\partial f(Y^*)}{\partial \mathbf{A}} = 2(\underline{\mathbf{u_1}\ \mathbf{u}_1'\mathbf{A}} - \underline{\mathbf{u_2}\ \mathbf{u}_2'\mathbf{A}}) = 2(\boxed{\mathbf{u_1}\delta_1\mathbf{v}_1' - \mathbf{u_2}\delta_2\mathbf{v}_2'})$$

rank-2 SVD

    - Avoid matrix multiplication ($\mathrm{O}(n^3)$ in time).
    - Construct while optimizing HITS.
- **Time and Space Complexities:**
  - $\mathrm{O}(m+n)$ in time and $\mathrm{O}(m+n)$ in space.

[1] Ng, A. Y., Zheng, A. X., & Jordan, M. I. Stable algorithms for link analysis. SIGIR 2001.

Task #2: Spectral Clustering

- **Goal:** find k clusters such that $\begin{cases} \text{maximize intra-connectivity} \\ \text{minimize inter-connectivity} \end{cases}$
- **Optimization Problem:**

$$\begin{aligned} \min_{\mathbf{U}} \quad & \mathrm{Tr}\left(\mathbf{U}'\mathbf{L}\mathbf{U}\right) \\ \mathrm{s.t.} \quad & \mathbf{U}'\mathbf{U} = \mathbf{I} \end{aligned}$$

  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
- **Question:** what would happen if an edge is perturbed between two nodes?

N2N for Spectral Clustering

- **Choice of $f(\cdot)$ Function:** $f(Y^*) = \mathrm{Tr}(\mathbf{U}'\mathbf{L}\mathbf{U})$
  - Consistent with the objective.
- **Constructing the Derivative Network:**
  - **Chain Rule:**

$$\frac{\partial f(Y^*)}{\partial \mathbf{A}(i,j)} = \mathrm{Tr}\left[\left(\frac{\partial \sum_{i=1}^{k} \lambda_i}{\partial \mathbf{L}}\right)' \frac{\partial \mathbf{L}}{\partial \mathbf{A}(i,j)}\right] = \mathbf{U}'(i,:)[\mathbf{U}(i,:) - \mathbf{U}(j,:)]$$

  - **Matrix Form Solution:**

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

$\mathbf{1}_{n\times n}$: $n\times n$ full matrix with 1 as entries.

    - **Limitation:** $O(n^3)$ in time complexity and $O(n^2)$ in space.
    - **Question:** how to scale up the computation to large networks?

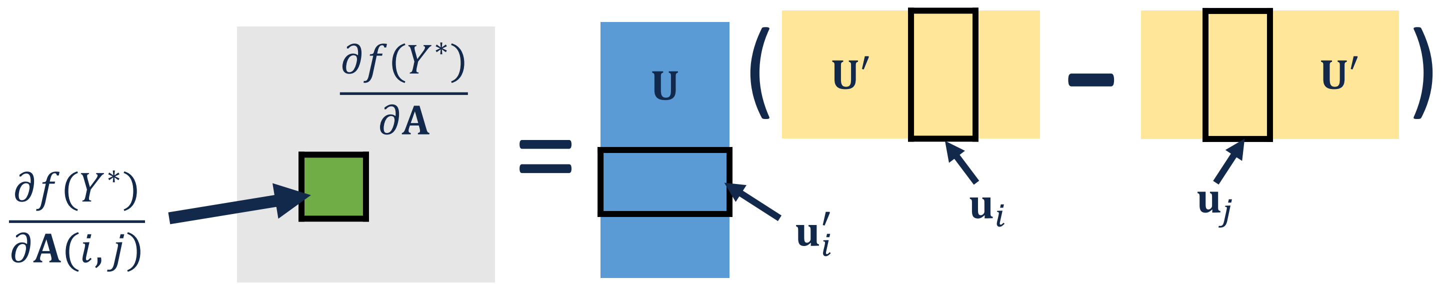
N2N for Spectral Clustering

- **Scale-Up Computation:** $\frac{\partial f(Y^*)}{\partial \mathbf{A}} = \mathrm{diag}(\mathbf{U}\mathbf{U}')\mathbf{1}_{n\times n} - \mathbf{U}\mathbf{U}'$
  - **Solution:** explore the low-rank structure.

$$\mathrm{diag}(\mathbf{U}\mathbf{U}')\mathbf{1}_{n\times n} = \mathrm{diag}(\mathbf{U}\mathbf{U}')\mathbf{1}_{n\times 1}\mathbf{1}_{1\times n} = \begin{bmatrix} \mathbf{u}_1'\mathbf{u}_1 \\ \dots \\ \mathbf{u}_n'\mathbf{u}_n \end{bmatrix} \mathbf{1}_{1\times n} \text{ and } \mathbf{U}\mathbf{U}' = \begin{bmatrix} \mathbf{u}_1' \\ \dots \\ \mathbf{u}_n' \end{bmatrix} [\mathbf{u}_1 \ \dots \mathbf{u}_n]$$

$\mathbf{u}_i'$ = i^{th} row of matrix $\mathbf{U}$

  - $\frac{\partial f(Y^*)}{\partial \mathbf{A}(i,j)} = \mathbf{u}_i'(\mathbf{u}_i - \mathbf{u}_j) \rightarrow O(k)$ time for one edge and $O(km)$ in total.



- **Time and Space Complexities:**
  - $O(k(m+n) + k^2 n)$ in time and $O(kn+m)$ in space.

Task #3: Matrix Completion

- **Goal:** learn two low-rank matrices for all users and items
- **Optimization Problem:**

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

Ω = {observations}, λ_u, λ_u for regularization

not jointly convex for $\mathbf{U}$ and $\mathbf{V}$

- **Solution:** Alternating Least Square (ALS)
  - Fix $\mathbf{U}$, solve for $\mathbf{V}$
  - Fix $\mathbf{V}$, solve for $\mathbf{U}$
- **Question:** how sensitive are the recommendation results due to users' ratings?

N2N for Matrix Completion

- **Choice of $\boldsymbol{f}(\cdot)$ Function:** $f(Y^*) = \|\mathbf{UV}'\|_F^2$
  - Measure the overall sensitivity of the recommender system.
- **Constructing the Derivative Network:**
  - **Chain Rule: $\mathbf{X} = \mathbf{UV}'$**

$$\frac{\partial f(Y^*)}{\partial \mathbf{A}(i,j)} = \sum_{l}^{n_1} \sum_{t}^{n_2} \frac{\partial f(Y^*)}{\partial \mathbf{X}(l,t)} \frac{\partial \mathbf{X}(l,t)}{\partial \mathbf{A}(i,j)}$$

$$= 2 \sum_{l}^{n_1} \sum_{t}^{n_2} \mathbf{X}(l,t) \left[\boxed{\frac{\partial \mathbf{U}(l,:)}{\partial \mathbf{A}(i,j)}} \mathbf{V}'(t,:) + \mathbf{U}'(l,:) \boxed{\frac{\partial \mathbf{V}(t,:)}{\partial \mathbf{A}(i,j)}} \right]$$

involve optimization procedure

  - **Observation:** optimization procedure is involved.
  - **Solution:** consider KKT conditions [Li et al. 2016]

[1] Li, B., Wang, Y., Singh, A., & Vorobeychik Y.. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. NIPS 2016.

N2N for Matrix Completion

- **Solution:** consider KKT conditions.
  - $\frac{\partial \mathbf{U}(l,:)}{\partial \mathbf{A}(i,j)} = \mathbf{V}(j,:)[\lambda_u \mathbf{I} + \sum_{k\in\Omega_l} \mathbf{V}(k,:)'\mathbf{V}(k,:)]$, only if $i = l$
  - $\frac{\partial \mathbf{V}(t,:)}{\partial \mathbf{A}(i,j)} = \mathbf{U}(i,:)[\lambda_v \mathbf{I} + \sum_{k\in\Omega_t} \mathbf{U}(k,:)'\mathbf{U}(k,:)]$, only if $j = t$
  - 0 otherwise
- **Element-wise Solution:** $\mathbf{C}_i = \lambda_u \mathbf{I} + \sum_{k\in\Omega_i} \mathbf{V}(k,:)'\mathbf{V}(k,:)$, $\mathbf{D}_\mathrm{j} = \lambda_v \mathbf{I} + \sum_{k\in\Omega_j} \mathbf{U}(k,:)'\mathbf{U}(k,:)$

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

many matrix multiplications rating-specific terms

  - **Observation:**
    - need to calculate $\mathbf{C}_i$ and $\mathbf{D}_j$ for each rating made by a user.
    - many matrix multiplications involved.
  - **Question:** how to scale up to large networks?

N2N for Matrix Completion

- **Element-wise Solution:**

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

- **Scale-up computation:**
  - $\mathbf{U}$, $\mathbf{V}$**:** mining results.
  - $\mathbf{U}'\mathbf{U}$, $\mathbf{V}'\mathbf{V}$: shared for all users and items.
  - $\mathbf{C}_i$, $\mathbf{D}_j$: calculated during ALS.
  - Precompute $\mathbf{U}'\mathbf{U}$, $\mathbf{V}'\mathbf{V}$, $\mathbf{C}_i$ and $\mathbf{D}_j$ during ALS.
- **Time and Space Complexities:**
  - $\mathrm{O}(k^3(n_1+n_2)+k^2m)$ amortized time complexity
  - $\mathrm{O}(k^2(n_1+n_2)+m)$ space complexity.

Roadmap

- Motivations ✓
- N2N Instantiations ✓
  - Task #1: HITS
  - Task #2: Spectral Clustering
  - Task #3: Matrix Completion
- Experimental Results
- Conclusions

Experimental Settings

- **Questions:**
  - Effectiveness in attacking network mining tasks
  - Scalability
- **Datasets:** 10+ various types of real-world datasets.
  - **Types:** directed, undirected, bipartite
- **Baseline Methods:**

Method	Parameter
Top Degree	-
Top EigenCentrality	-
HITS	-
CONTAIN[1]	$r = 80$
AURORA[2]	$c = 1/2\lambda_{\max}(\mathbf{A})$

[1] Chen, C., Peng R., Ying L., & Tong, H. Network Connectivity Optimization: Fundamental Limits and Effective Algorithms. KDD 2018.
[2] Kang, J., Wang, M., Cao, N., Xia, Y., Fan, W., & Tong, H. AURORA: Auditing PageRank on Large Graphs. Big Data 2018.

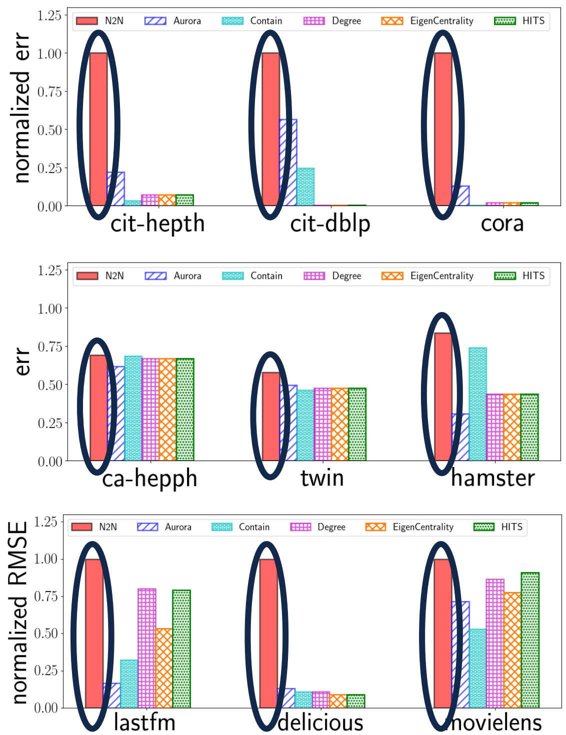
Effectiveness

- **Observation:** outperform baseline methods across three different instantiations and over different budget size.

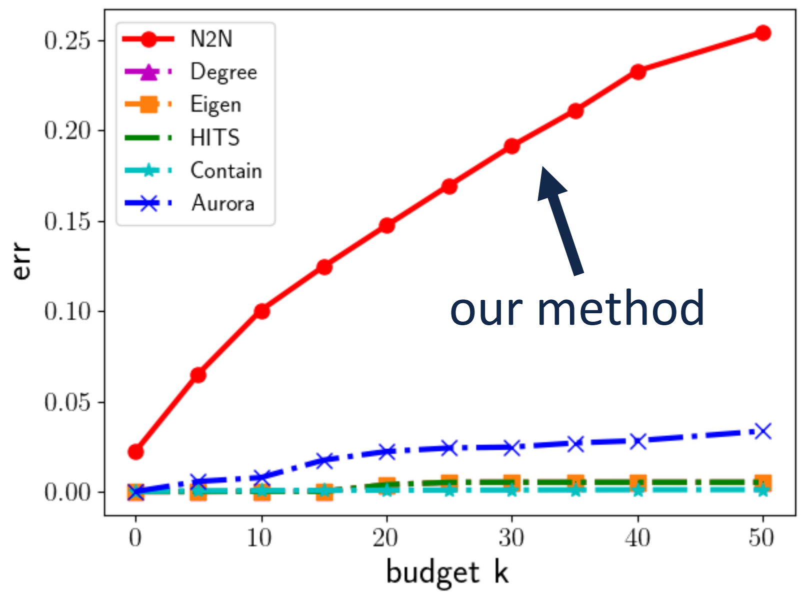
Ranking by HITS

Spectral Clustering

Matrix Completion



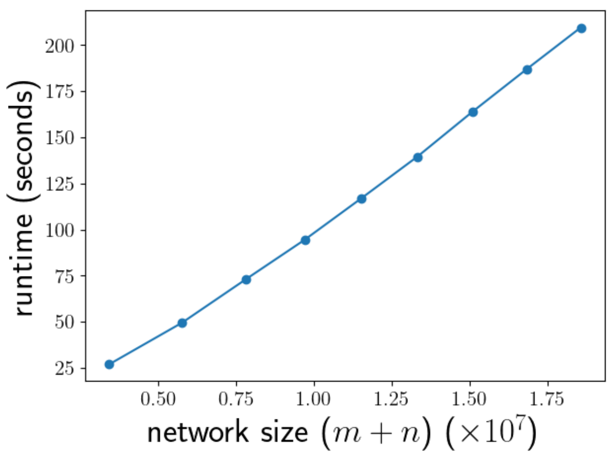
Ranking by HITS



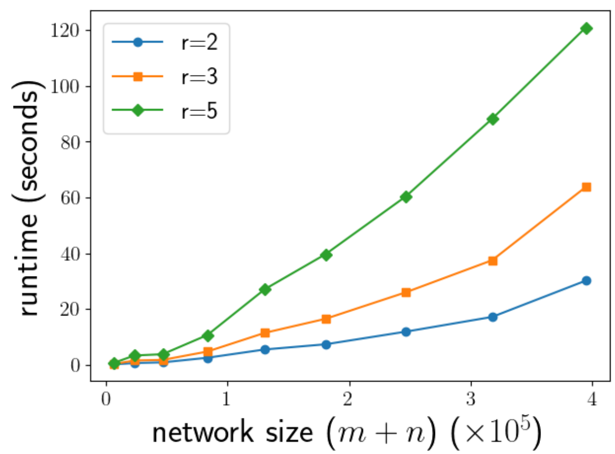
Scalability

- **Observation:** scale linearly w.r.t. network size.

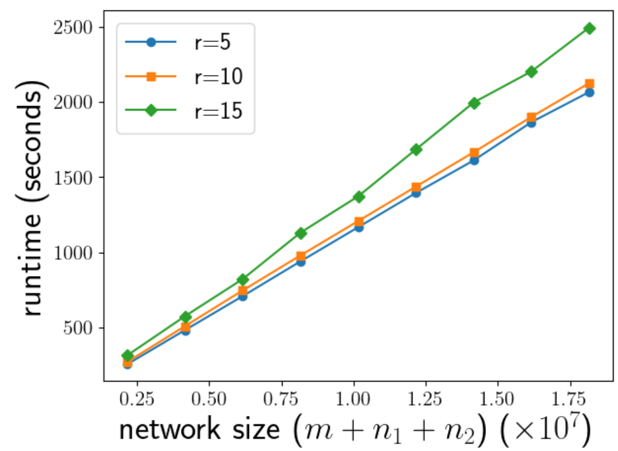
Ranking by HITS



Spectral Clustering



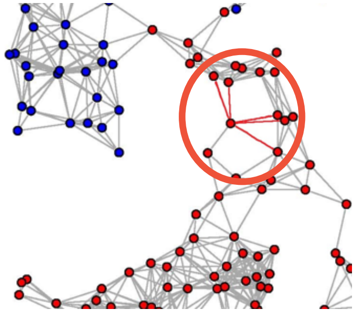
Matrix Completion

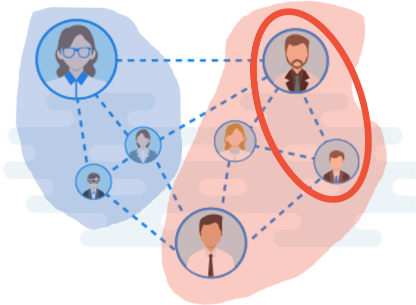


Roadmap

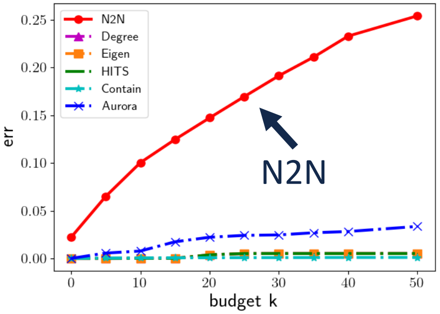
- Motivations ☑
- N2N Instantiations ☑
  - Task #1: HITS
  - Task #2: Spectral Clustering
  - Task #3: Matrix Completion
- Experimental Results ☑
- Conclusions

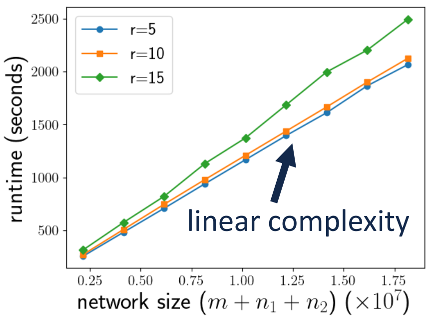
Conclusions





- **Problem:** Network Derivative Mining
- **Solution:**
  - An algorithmic framework (N2N) to construct the derivative network.
    - Three different instantiations: HITS, spectral clustering, matrix completion
  - Ability to scale up to large networks.
- **Results:**
  - Effective in adversarial attack on network mining tasks.
  - Linear time and space complexity.





- More details in the paper.
  - Detailed experimental settings.
  - Additional experimental results.
  - Proofs and analysis for each instantiation.

A Special Thank to SIGIR

- I would like to thank SIGIR for offering me the travel award to attend this prestigious conference.