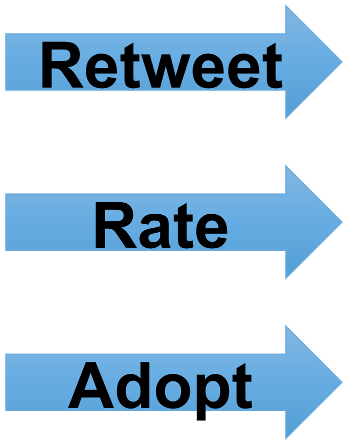
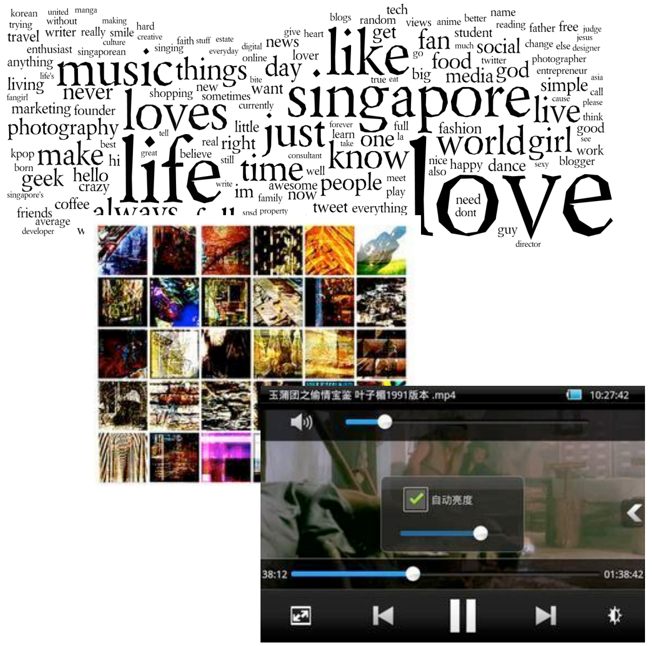
KDD’14 Tutorial
Network Mining and Analysis for Social Applications

Feida Zhu
Singapore Management University
Huan Sun,   Xifeng Yan
University of California at Santa Barbara

August 24, 2014

Our perspective: social applications





Friendship Network

Collective behavior

⋮

User

- Follow
- Befriend
- …

Feedback

- Retweet
- Reply
- Rate
- Adopt
- …

Content

- Usage of hashtags
- Usage of lexicons
- …

Our perspective: social applications

- The essence of “Social Data”--- what makes it different from traditional relational table data
    - Interactive behavior
        - Behavioral patterns, anomalies, collaborations, relationship, alignment, etc.
    - Network Structure
        - Structural properties, measures, patterns, correlation, evolution, etc.
- This tutorial examines network mining and analysis from these two aspects as motivated by real social applications.

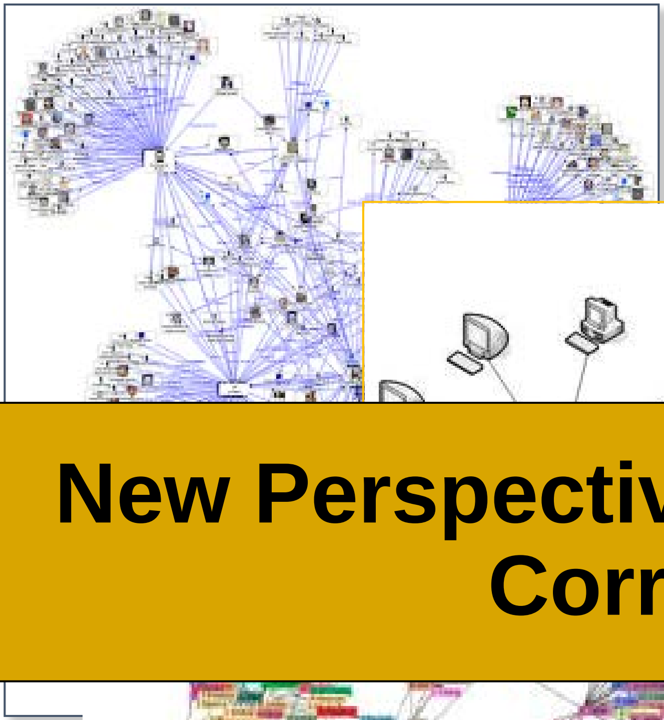
Tutorial Coverage

- Network Structure
  - Part I: Network Correlation and Patterns (Huan)
  - Part II: Frequent Network Patterns (Feida)
- Interactive Behavior
  - Part III: Collaboration Patterns (Huan)
- Network Structure + Interactive Behavior
  - Part IV: Relationship Mining (Feida)
  - Part V: User Identity Linkage (Feida)

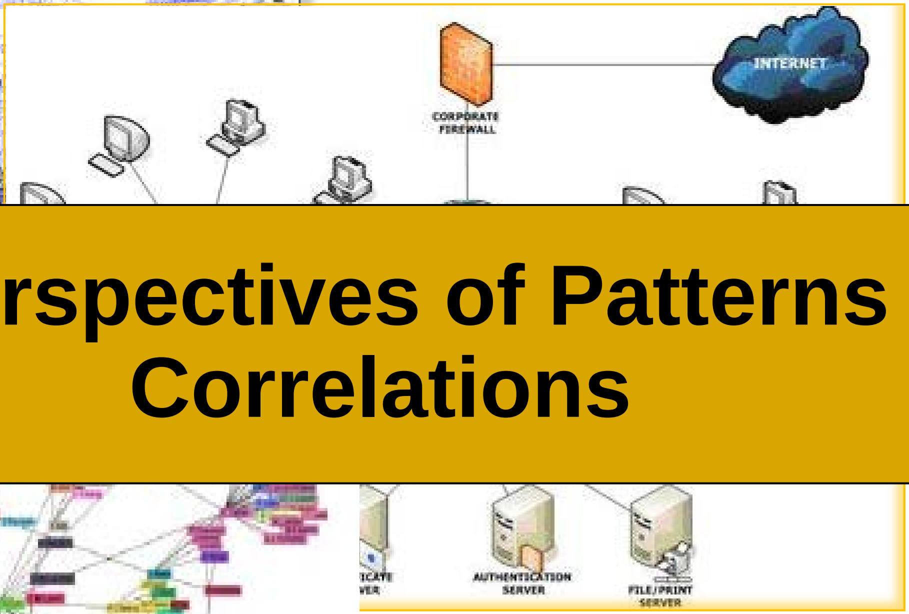
Network Mining and Analysis for Social Applications

Part I
Network Correlation and Patterns

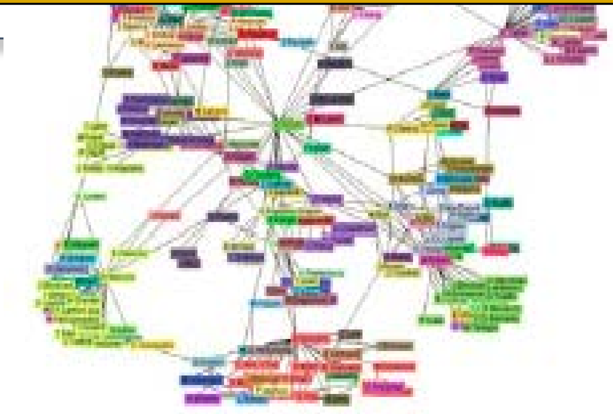
Networks with Rich Attributes



Social networks

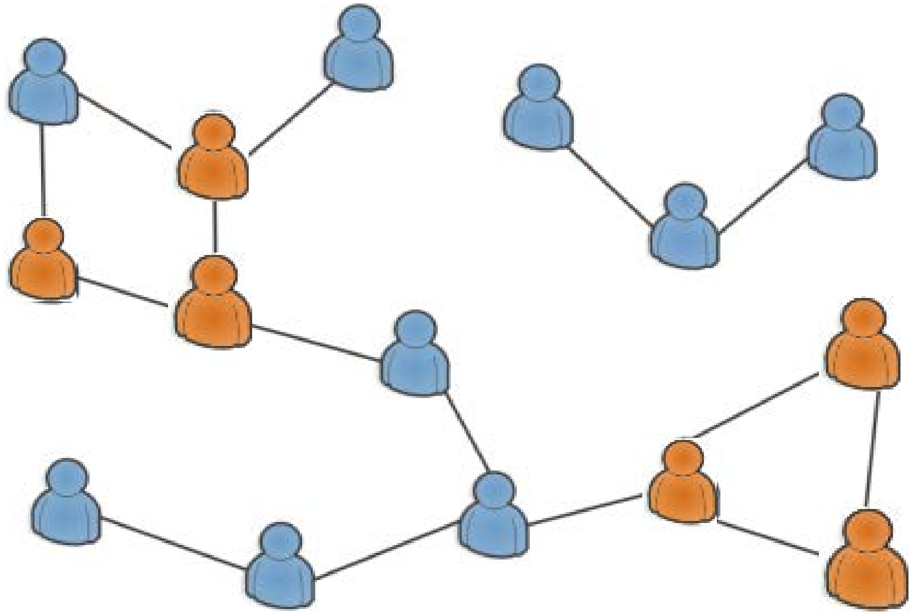


New Perspectives of Patterns and Correlations



Research Collaboration Networks

Example of Correlations



Correlation between the occurrence of an event and the network structure

Further study:

1. Is this correlation due to influence in the network? (network structure→ event occurrence)

2. Conversely, is it the preferences over the same product facilitate the formation of the link between users, i.e., their friendship development? (event occurrence → network structure)

Pattern Kaleidoscope

- ❑ Proximity Pattern
- ❑ Attribute-Structure Correlations
- ❑ Cohesive Pattern
- ❑ Itemset-sharing Pattern
- ❑ Graph Topological pattern
- ❑ Graph Iceberg
- ❑ Graph Anomaly
- ❑ Frequent Network Pattern

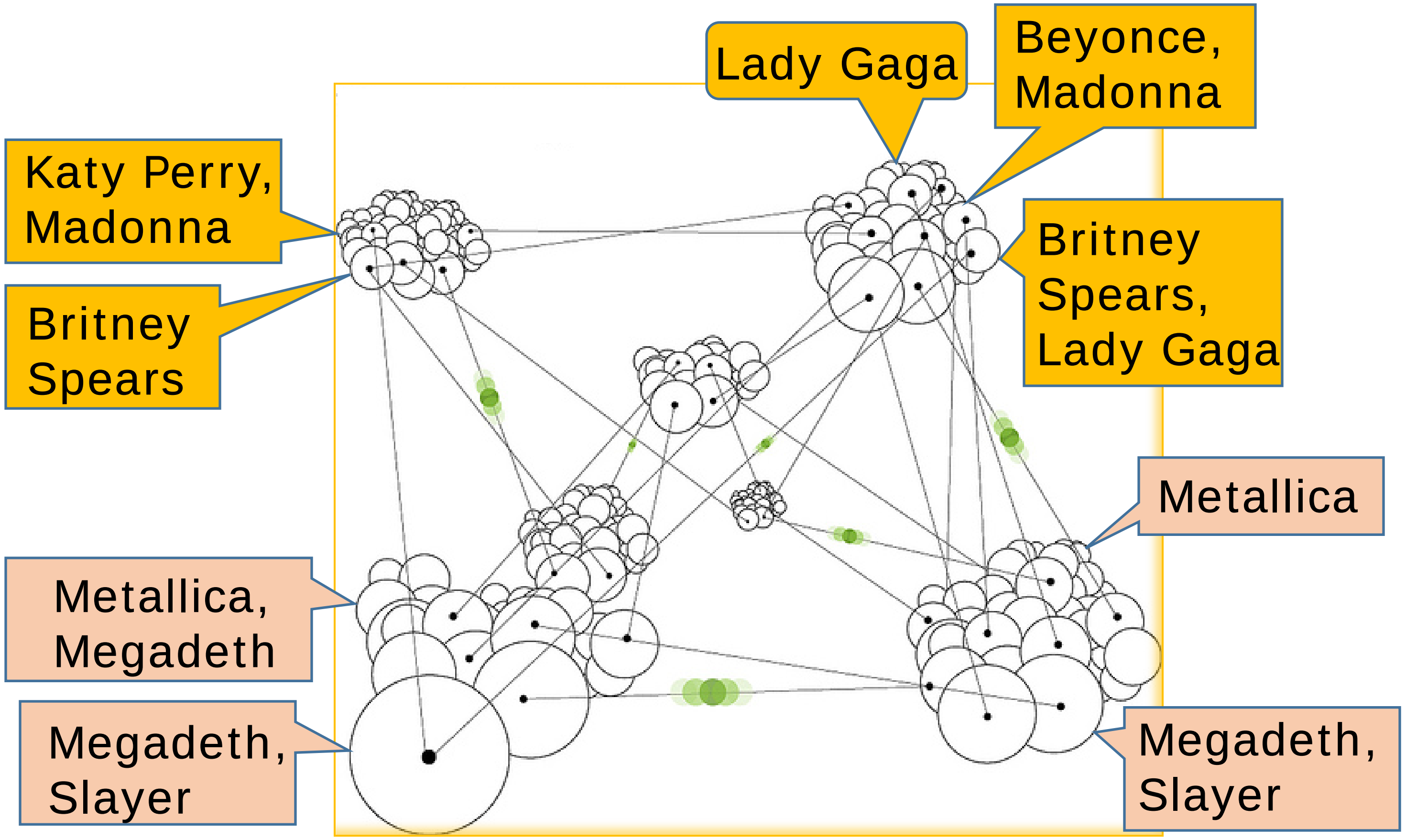
Pattern Kaleidoscope

- ❑ Proximity Pattern ✓
- ❑ Attribute-Structure Correlations ✓
- ❑ Cohesive Pattern
- ❑ Itemset-sharing Pattern
- ❑ Graph Topological Pattern

$\Big\}$ **Brief introduction**

- ❑ Graph Iceberg ✓
- ❑ Graph Anomaly ➡ **Akoglu et al., Tutorial at WSDM'13**
- ❑ Frequent Network Pattern ✓ (Part II)

What are the **related Musical Bands/ Singers** that **co-occur frequently in neighborhood?**



Social Network

Last.FM

Nodes -> Users

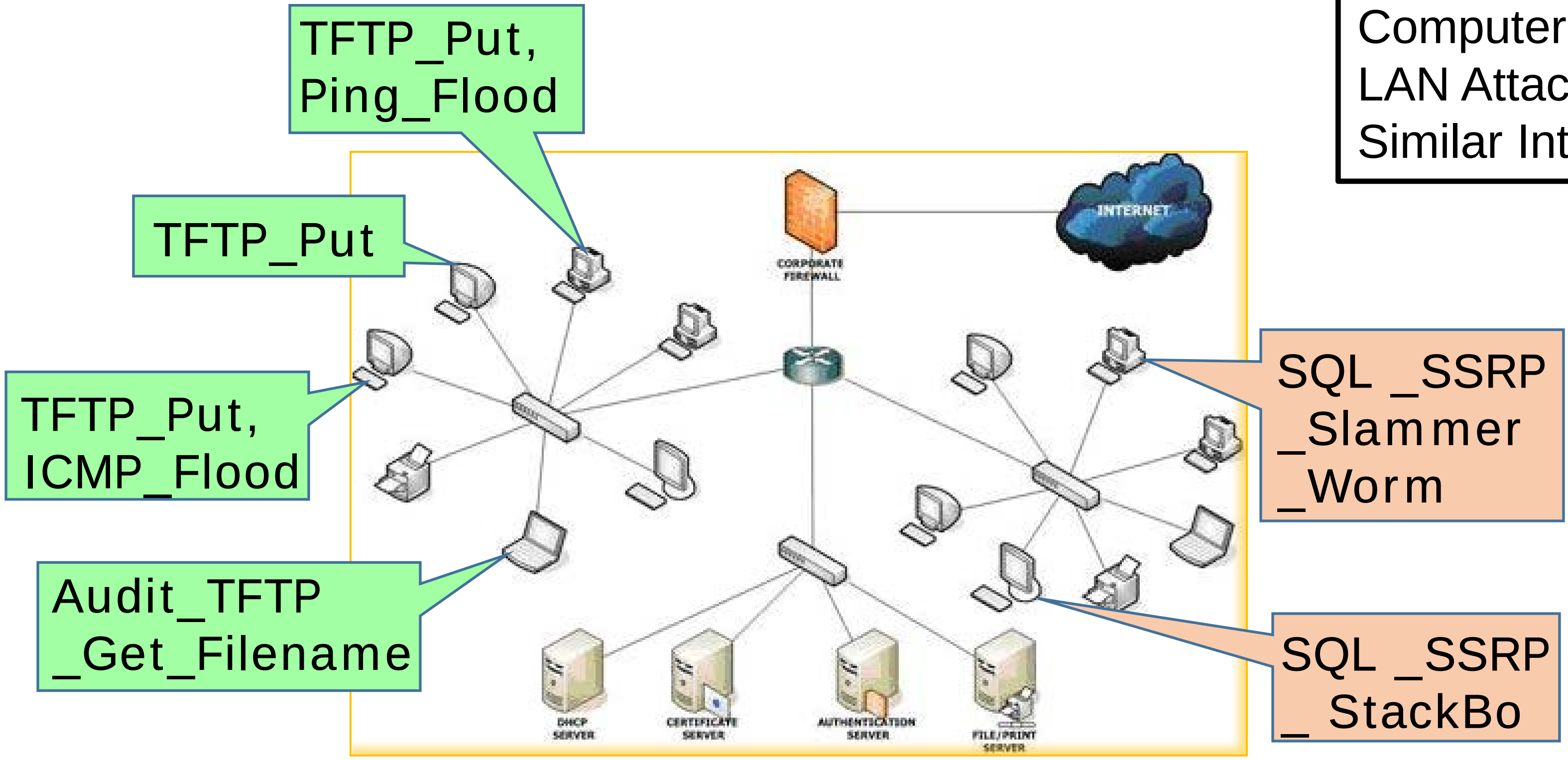
Edges -> Links

List of Musical Bands/ Singers

What are **Related Computer Attacks** that **Co-occur Frequently in Neighborhood?**

Computers in LAN

Computers in Same LAN Attacked by Similar Intrusions



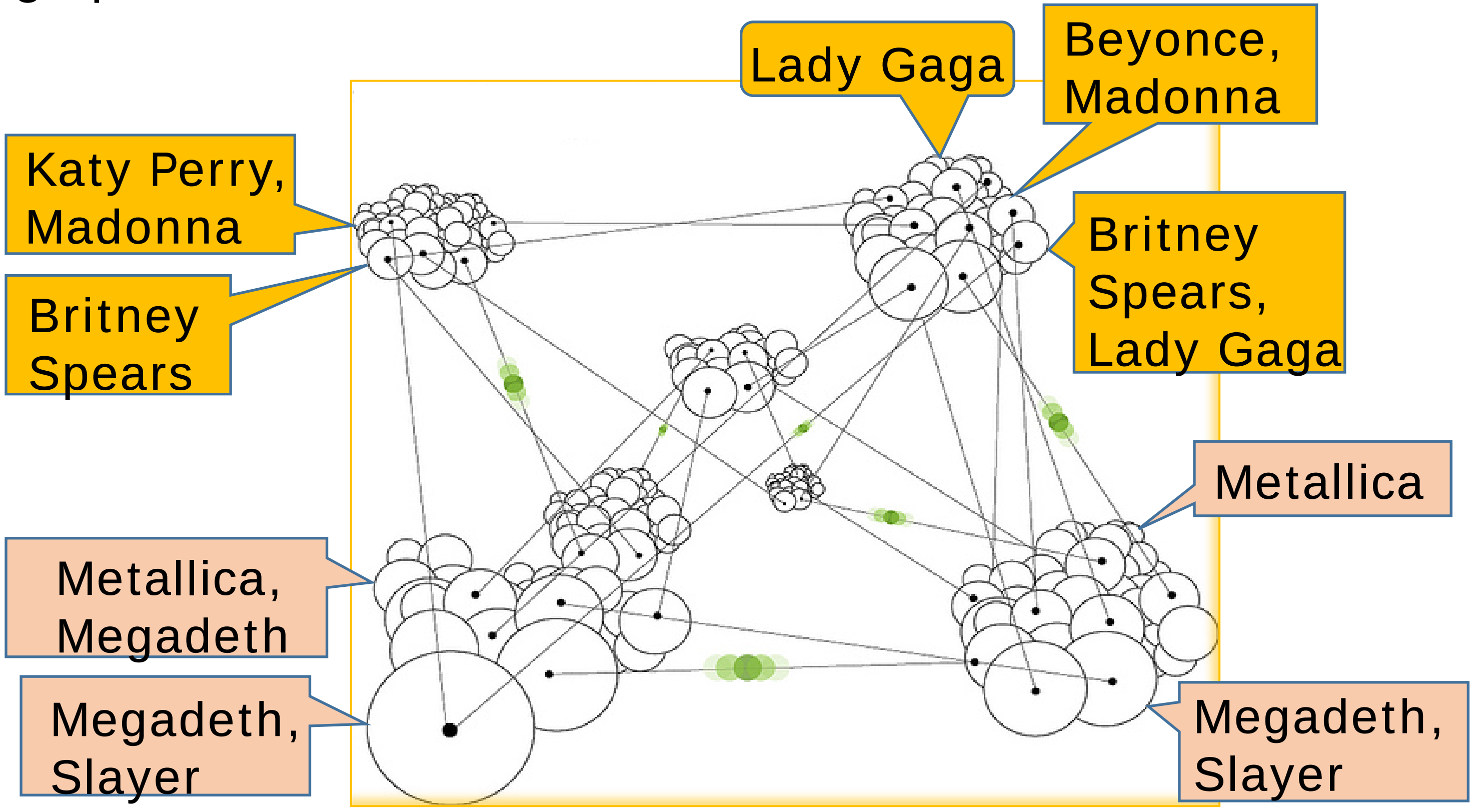
Intrusion Network

[Khan et al., SIGMOD'10]

Proximity Pattern Mining

❑ Definition

A subset of labels that repeatedly appear in tightly connected subgraphs in G.

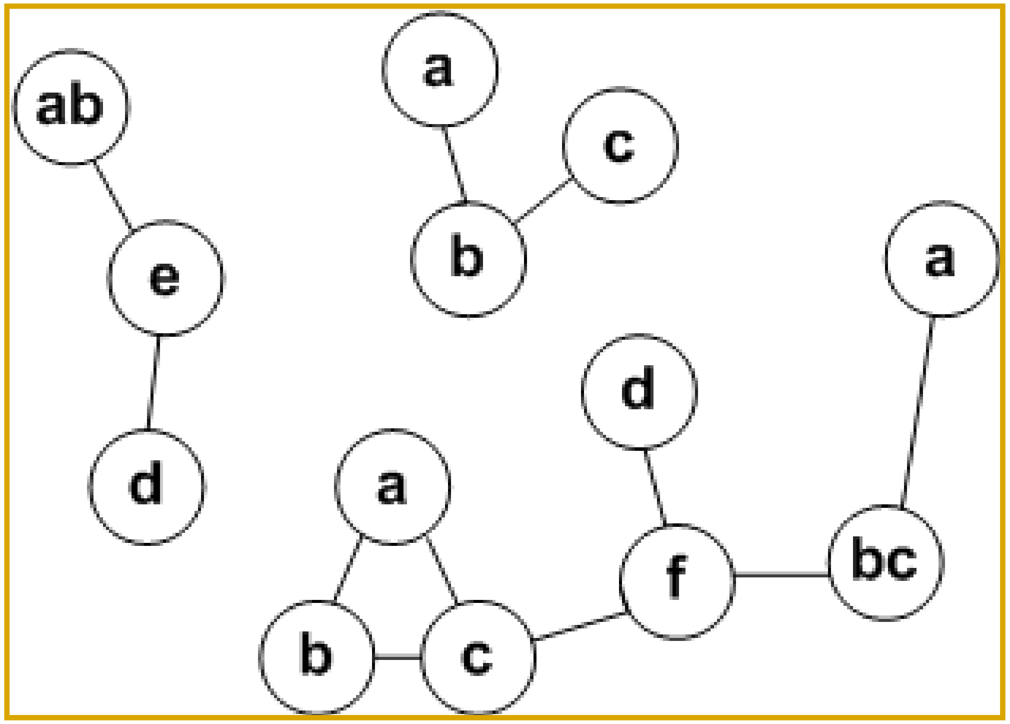


[Khan et al., SIGMOD'10]

Proximity Pattern Mining

❑ Definition

A subset of labels that repeatedly appear in tightly connected subgraphs in G.



a, b – YES
a, b, c – YES
d, e, f - NO

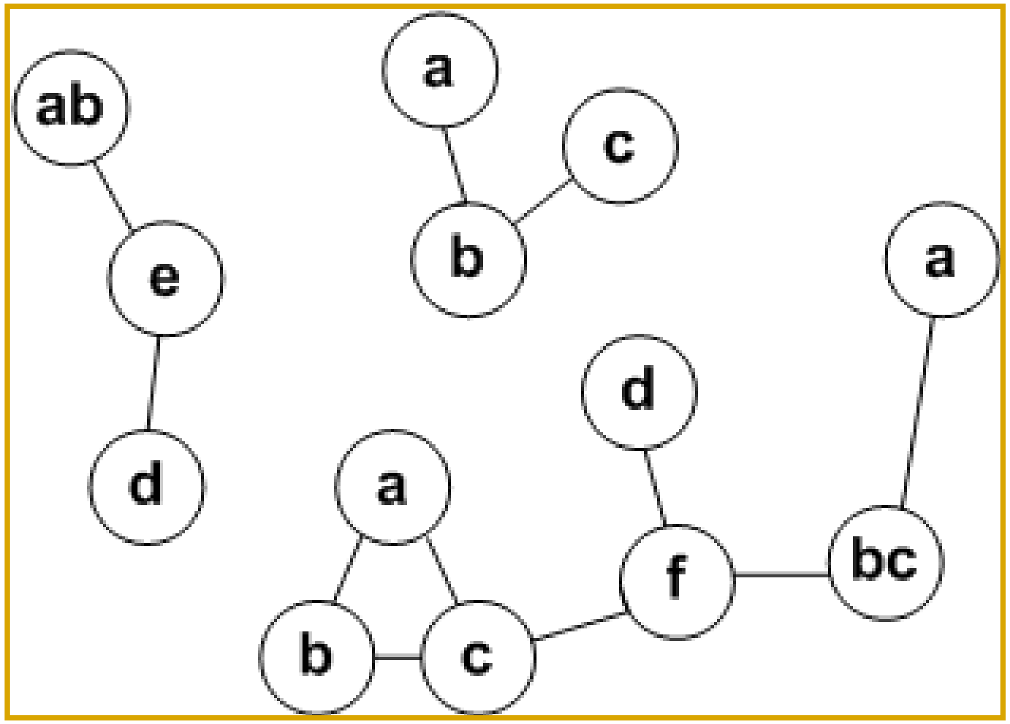
Proximity Pattern Definition

- **Characteristics**

  (1) Proximity

  (2) Frequency

  (3) Flexibility

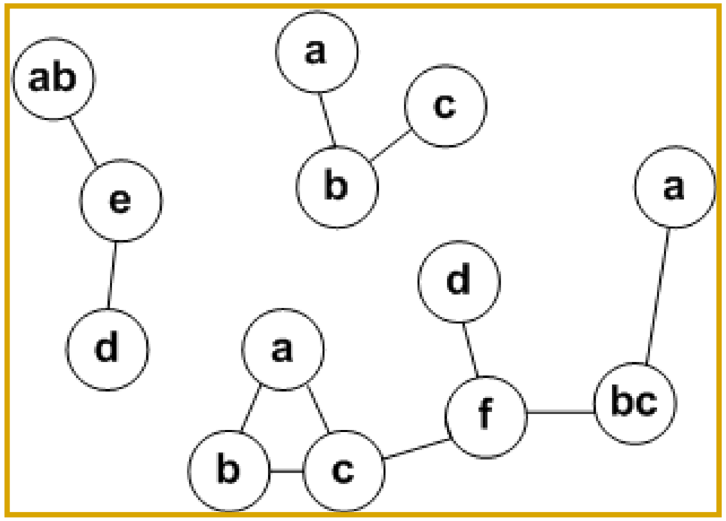


a, b – YES; a, b, c – YES
d, e, f - NO

[Khan et al., SIGMOD’10]

Proximity Pattern Definition

- Will Frequent Subgraph Mining Work?

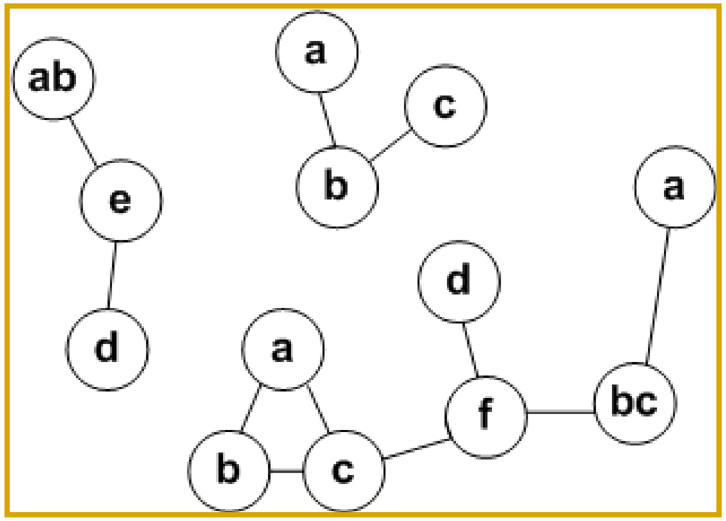


Proximity pattern: {a, b, c}

[Khan et al., SIGMOD'10]

Proximity Pattern Definition

- ❑ Will Frequent Subgraph Mining Work?
  - NO !!!
  - **Lack of Flexibility**

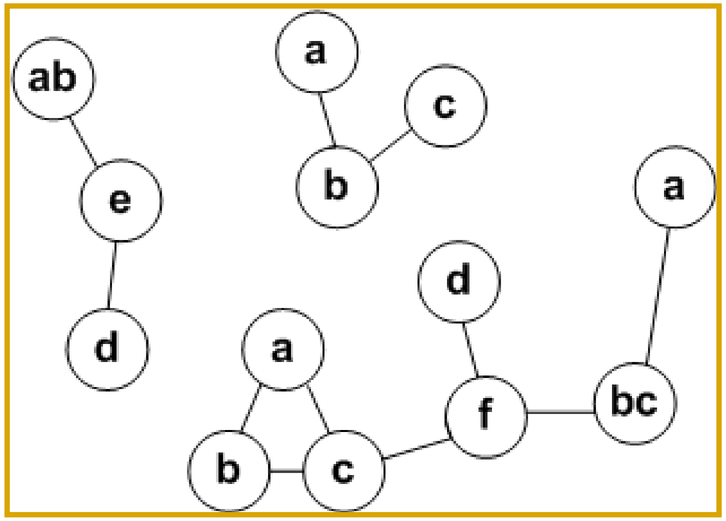


Proximity pattern: {a, b, c}

[Khan et al., SIGMOD'10]

Proximity Pattern Definition

- Will Frequent Subgraph Mining Work?
  - NO !!!
  - Lack of Flexibility
- Will Frequent Itemset Mining Work?

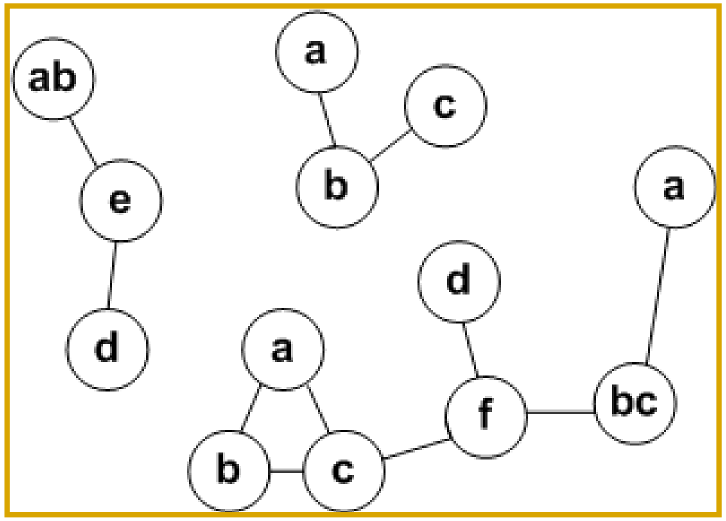


Proximity pattern: {a, b, c}

[Khan et al., SIGMOD'10]

Proximity Pattern Definition

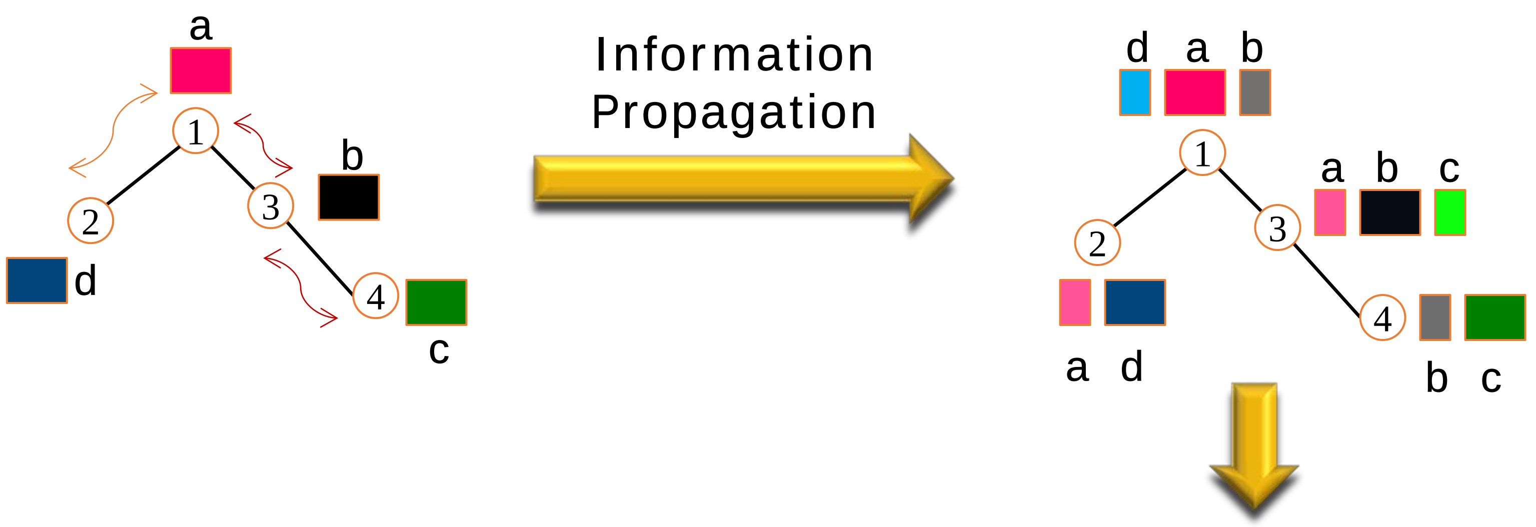
- Will Frequent Subgraph Mining Work?
  - NO !!!
  - Lack of Flexibility
- Will Frequent Itemset Mining Work?
  - NO !!!
  - No Notion of Edge



Proximity pattern: {a, b, c}

[Khan et al., SIGMOD'10]

Information Propagation Model

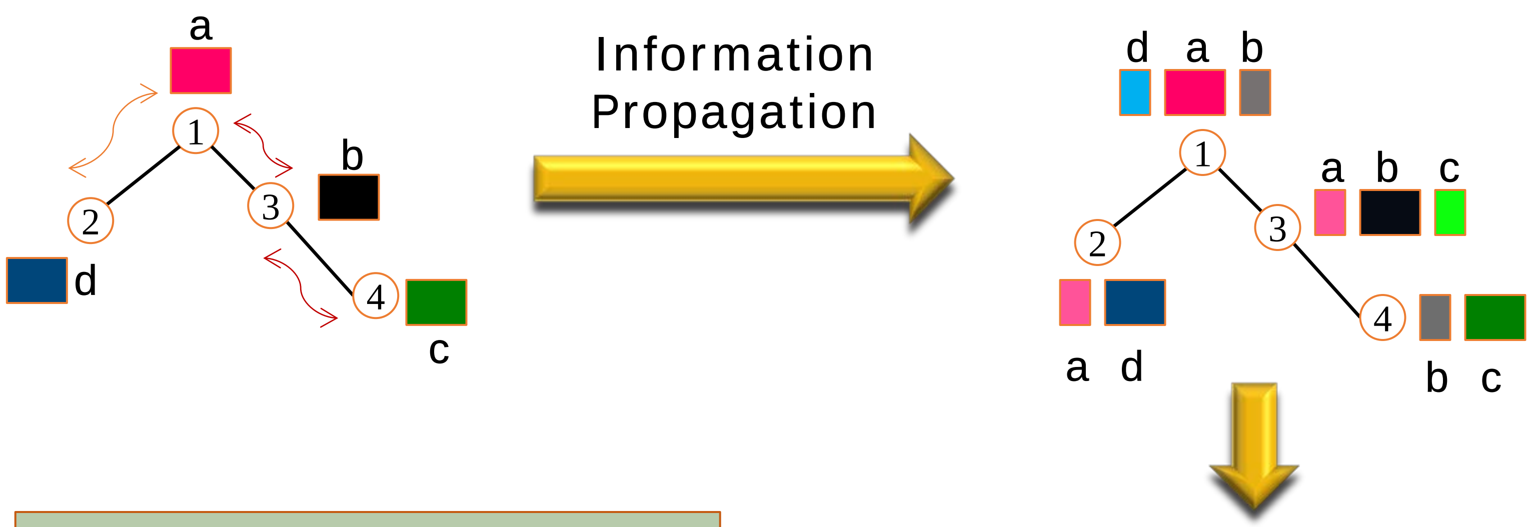


	a	b	c	d
1	1.00	0.12	0.00	0.12
2	0.19	0.00	0.00	1.00
3	0.12	1.00	0.12	0.00
4	0.00	0.19	1.00	0.00

UCSB SMU SINGAPORE MANAGEMENT UNIVERSITY

[Khan et al., SIGMOD'10]

Information Propagation Model



- **Frequent-Pattern (FP) Tree** cannot handle fractional association values because of the new definition of Support.
- Modify FP Tree Structure and Algorithm.
- *C. C. Aggarwal et. al (KDD '09), Bernecker et. al (KDD '09).*

	a	**b**	**c**	**d**
1	1.00	0.12	0.00	0.12
2	0.19	0.00	0.00	1.00
3	0.12	1.00	0.12	0.00
4	0.00	0.19	1.00	0.00

Frequent Itemset Mining (Probabilistic)

[Khan et al., SIGMOD'10]

Top-k Interesting Patterns

- How to measure "Interesting-ness"?

----Randomization Test

Generate graph *Q* from graph *G* by randomly swapping the labels among nodes. Let, *p* and *q* be the support values of itemset *I* in *G* and *Q* respectively. High difference indicates interestingness.

G-test Score:

$$\boxed{p \cdot \ln \frac{p}{q} + (1-p) \cdot \ln \frac{1-p}{1-q}}$$

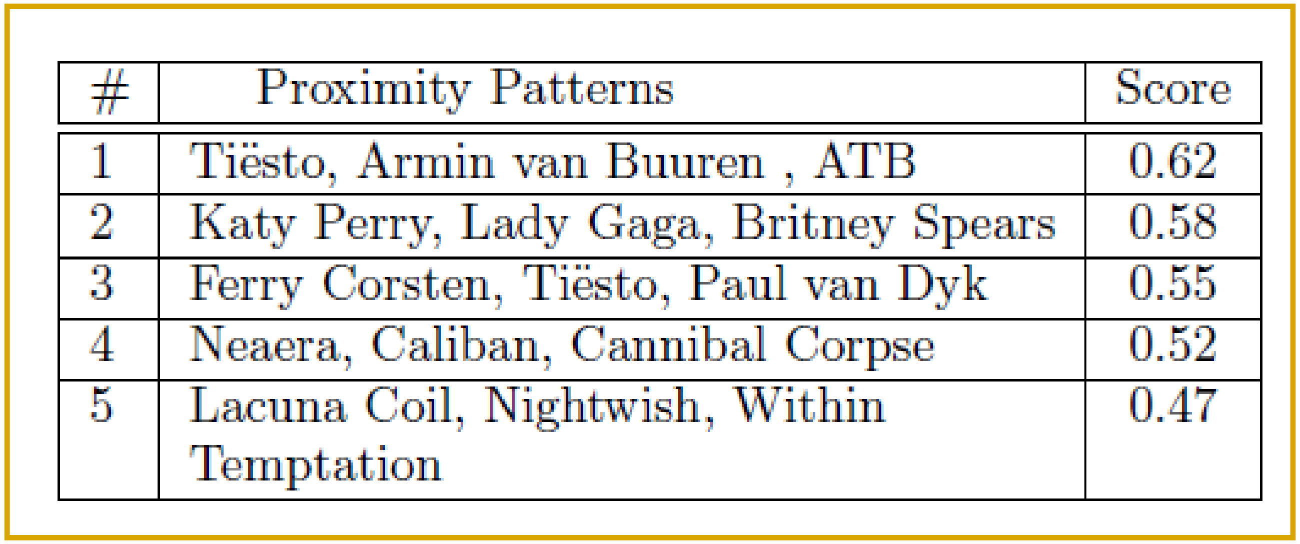
----Proximity Patterns minus Frequent Itemset Patterns

[Khan et al., SIGMOD'10]

Top-k Interesting Patterns

- Last.FM

Proximity Patterns



#	Proximity Patterns	Score
1	Tiësto, Armin van Buuren , ATB	0.62
2	Katy Perry, Lady Gaga, Britney Spears	0.58
3	Ferry Corsten, Tiësto, Paul van Dyk	0.55
4	Neaera, Caliban, Cannibal Corpse	0.52
5	Lacuna Coil, Nightwish, Within Temptation	0.47

- Britney Spears, Lady Gaga, Katy Perry – **American Female Pop Singers**
- ATB, Paul van Dyk – **German DJ**
- Tiesto, Ferry Corsten, Armin van Buuren – **Dutch DJ**
- Neaera, Caliban, Cannibal Corpse – **Death Metal Bands**
- Lucuna Coil, Nightwish, Within Temptation – **Gothic Metal Bands**

UCSB

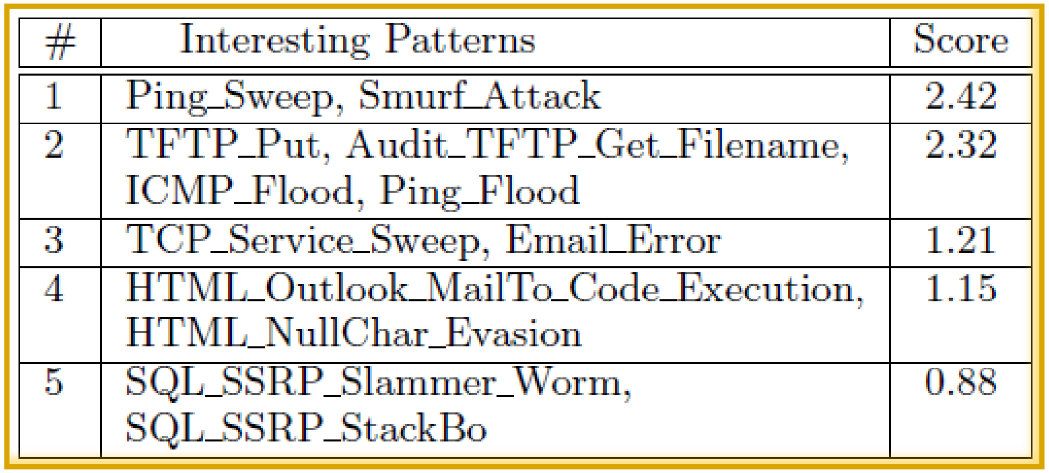
SMU SINGAPORE MANAGEMENT UNIVERSITY

[Khan et al., SIGMOD'10]

Top-k Interesting Patterns

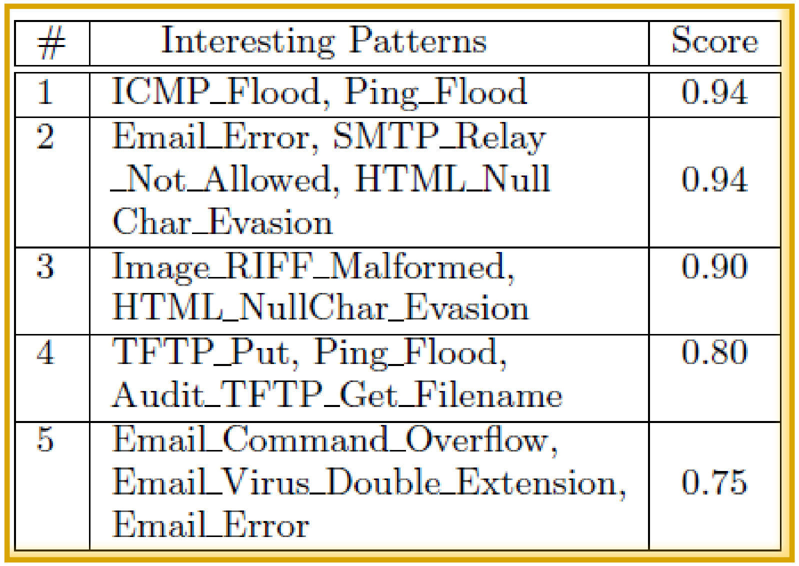
- Intrusion

Proximity Patterns



#	Interesting Patterns	Score
1	Ping_Sweep, Smurf_Attack	2.42
2	TFTP_Put, Audit_TFTP_Get_Filename, ICMP_Flood, Ping_Flood	2.32
3	TCP_Service_Sweep, Email_Error	1.21
4	HTML_Outlook_MailTo_Code_Execution, HTML_NullChar_Evasion	1.15
5	SQL_SSRP_Slammer_Worm, SQL_SSRP_StackBo	0.88

Proximity Patterns Minus Frequent Itemsets

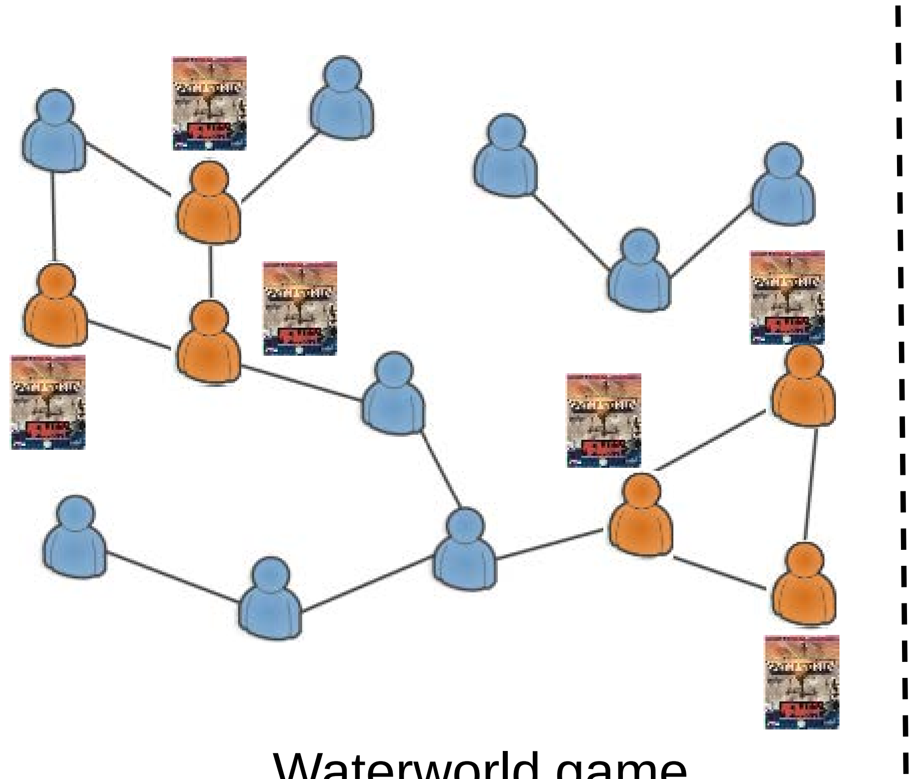


#	Interesting Patterns	Score
1	ICMP_Flood, Ping_Flood	0.94
2	Email_Error, SMTP_Relay _Not_Allowed, HTML_Null Char_Evasion	0.94
3	Image_RIFF_Malformed, HTML_NullChar_Evasion	0.90
4	TFTP_Put, Ping_Flood, Audit_TFTP_Get_Filename	0.80
5	Email_Command_Overflow, Email_Virus_Double_Extension, Email_Error	0.75

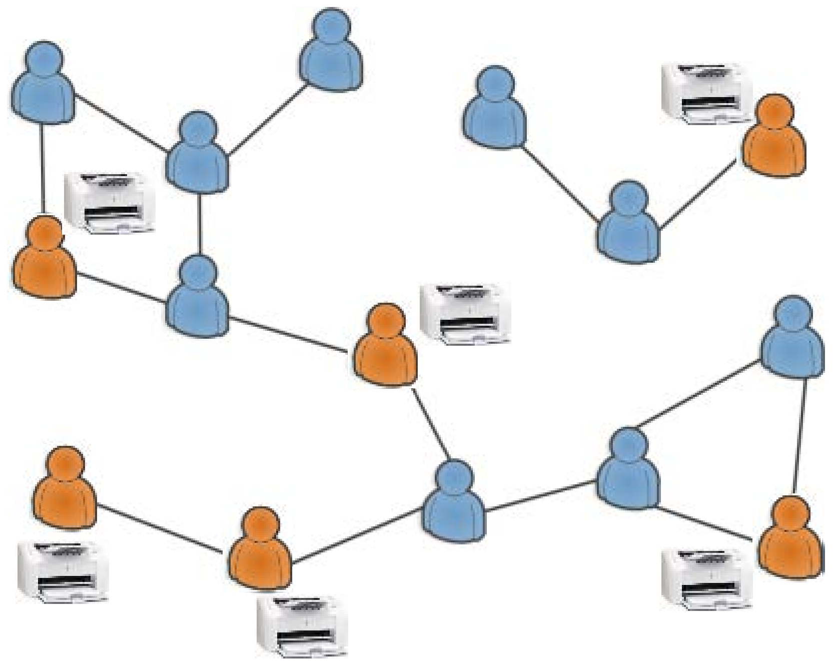
[Guan et al.,SIGMOD'11]

Structural Correlational Pattern

Which product's sales is more correlated with the social network structure?



Waterworld game

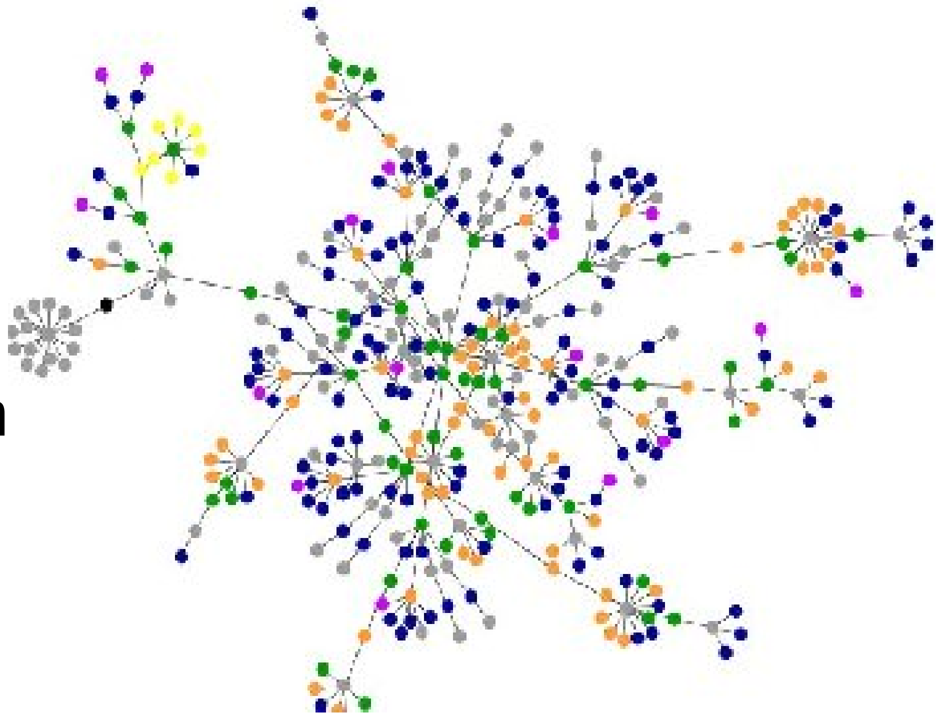


Hp printer

[Guan et al.,SIGMOD'11]

A General Situation

- ❑ Events taking place on nodes of a social graph
  - Online shopping
  - Blogging
  - Virus infection



- ❑ Social influence vs. Random occurrence

[Guan et al.,SIGMOD’11]

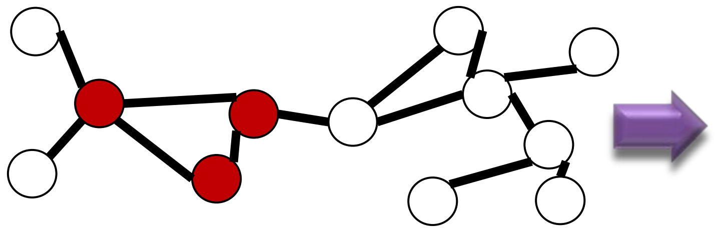
Why Measuring Such Correlations?

- ❑ Help understand the distribution of events in networks
- ❑ Help detect viral influence in the underlying network (e.g. product sales)
  - o Correlation has to do with link type, event type and time
  - o Facilitate product promotion, online ads recommendation

Problem Formulation

- A graph $G = (V, E)$ and an event set $Q = \{q_i\}$
- V_q--the set of nodes having event q. Let $|V_q| = m$, $|V| = n$

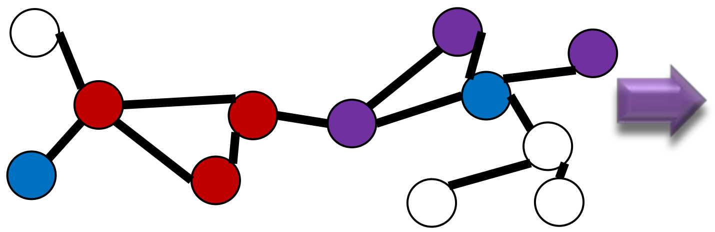
(1)



Is q1 correlated in G?

q1 🟥 q2 🟪 q3 🟦

(2)

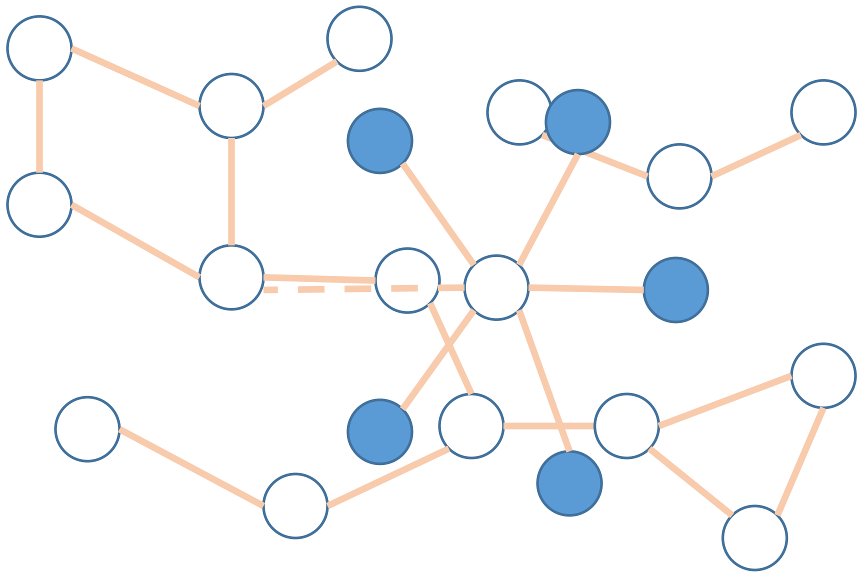


(1) q1
(2) q2
(3) q3

(ranking)

How to Characterize Correlation?

- If correlated, blue nodes tend to stick together.
- A naïve approach: only look at neighborhood
- General idea: compute the **aggregated proximity** among blue nodes



Measure Definition

- The measure is defined as

$$\rho(V_q) = \frac{\sum_{v \in V_q} s(v, V_q \setminus \{v\})}{|V_q|}$$

  - V_q: the set of nodes having event q; $s(\cdot)$ can be any graph proximity measure

- We choose **Hitting Time.**

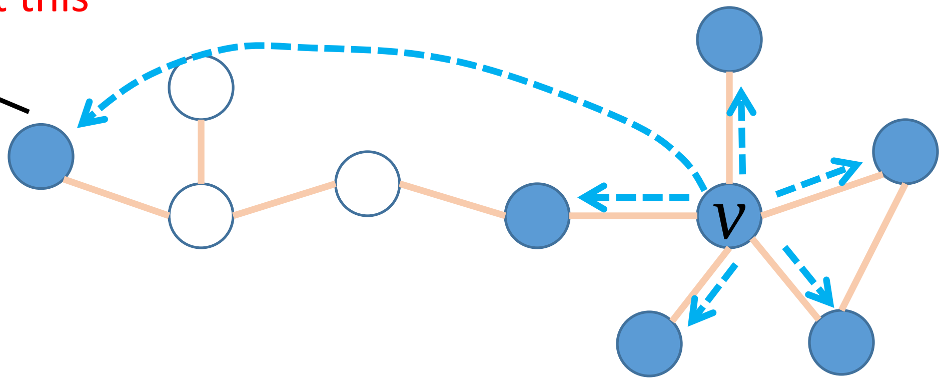
Measure Definition

- **Hitting time**: expected number of steps to reach a target node via random walk:

$$h(v_i, B) = \sum_{t=1}^{\infty} t \Pr(T_B = t \mid x_0 = v_i)$$

  - B: target node set; $\Pr(T_B=t|x_0=v_i)$: the probability that we start from v_i and reach B after t steps

Hitting time will not count this node



[Guan et al.,SIGMOD'11]

Hitting time & Decayed Hitting Time

- **Hitting time**: expected number of steps to reach a target node via random walk:

$$h(v_i, B) = \sum_{t=1}^{\infty} t \Pr(T_B = t \mid x_0 = v_i)$$

  - B: target node set; $\Pr(T_B{=}t|x_0{=}v_i)$: the probability that we start from v_i and reach B after t steps

Hitting time & Decayed Hitting Time

- **Hitting time**: expected number of steps to reach a target node via random walk:

$$h(v_i, B) = \sum_{t=1}^{\infty} t \Pr(T_B = t \mid x_0 = v_i)$$

  - B: target node set; $\Pr(T_B=t|x_0=v_i)$: the probability that we start from v_i and reach B after t steps

- **Decayed Hitting Time (DHT)**:

$$\tilde{h}(v_i, B) = \sum_{t=1}^{\infty} e^{-(t-1)} \Pr(T_B = t \mid x_0 = v_i)$$

  - Mapping [1,∞) to [0,1], high value means high proximity
  - Emphasizing the importance of local neighborhood and reducing the impact of long paths

- $\rho(V_q) = \frac{\sum_{v \in V_q} s(v, V_q \setminus \{v\})}{|V_q|}$, with $s(v, V_q \setminus \{v\})$ instantiated as $\tilde{h}(v, V_q \setminus \{v\})$

Measure Computation

- Directly compute $\rho(V_q)$ may be time consuming since $|V_q|$ may be large.

$$\rho(V_q) = \frac{\sum_{v \in V_q} s(v, V_q \setminus \{v\})}{|V_q|}$$

- Sampling: randomly select c nodes from V_q to estimate their DHTs to the remaining nodes.

[Guan et al.,SIGMOD’11]

Top-5 Structural Correlated Products (TaoBao)

Laptops and tablets

#	Product	Bound for $\tilde{\rho}$	m
1	ThinkPad T400	[554.43, 554,47]	47
2	Apple iPad	[227.56, 227.57]	698
3	ThinkPad X200	[91.39, 91.42]	60
4	Toshiba L600	[20.36, 20.41]	31
5	ThinkPad T410	[−1.13, −1.09]	72

Mobile and handheld devices

#	Product	Bound for $\tilde{\rho}$	m
1	iPod Touch 3	[92.06, 92.09]	484
2	Nokia 6300	[90.97, 90.99]	188
3	iPhone 4	[69.07, 69.09]	520
4	Nokia N82	[53.20, 53.24]	84
5	HTC G3	[36.48, 36.49]	732

Other

#	Product	Bound for $\tilde{\rho}$	m
1	Mamy Poko baby diapers	[238.50, 238.51]	4892
2	Beingmate Infant milk powder	[227.71, 227.72]	163
3	EVE game cards	[198.56, 198.58]	374
4	Mabinogi game cards	[189.56, 189.58]	446
5	Gerber cookies	[149.51, 149.52]	1491

Other (top 5 uncorrelated)

#	Product	Bound for $\tilde{\rho}$	m
1	Tiffany rings	[2.71, 2.72]	1092
2	Jack&Jones suits	[−0.48, −0.46]	311
3	Ray-Ban sunglasses	[−0.78,−0.77]	4958
4	Swarovski anklets	[−0.88,−0.84]	72
5	Jack&Jones shirts	[−3.28,−3.27]	1606

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

[Guan et al.,SIGMOD'11]

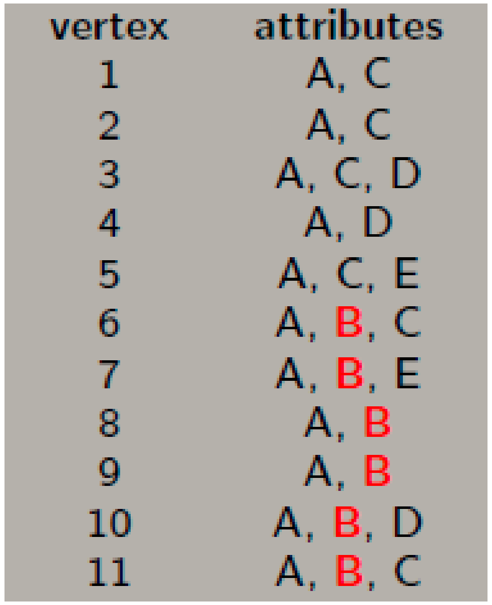
Top-12 Structural Correlated keywords (DBLP co-author network)

#	Keyword	Bound for $\tilde{\rho}$	m
1	Hadoop	[1225.22, 1225.46]	57
2	Microarray	[958.64, 958.67]	4738
3	OLTP	[912.52, 912.68]	105
4	AJAX	[857.61, 857.74]	179
5	Virus	[825.14, 825.21]	905
6	E-Learning	[811.85, 811.91]	3552
7	Database	[775.80, 775.83]	19522
8	Mining	[760.33, 760.36]	15371
9	Relational	[697.17, 697.22]	6225
10	Retrieval	[647.60, 647.64]	13996
11	Indexing	[624.94, 625.00]	5069
12	Computation	[586.58, 586.63]	11187

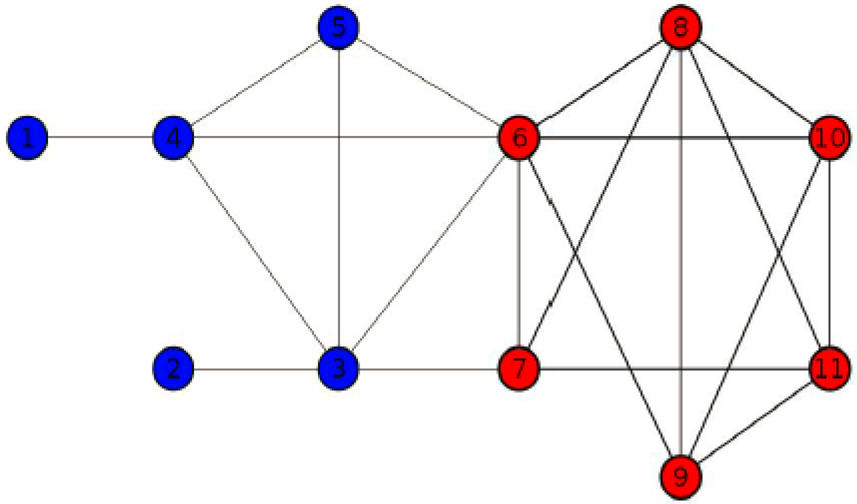
[Silva et al., VLDB'12]

Attribute-Structure Correlated Patterns

- Examples
  - o Densely connected webpages that share content
  - o Groups of friends with common interests
  - o Genes that interact and are expressed on the same issues.



vertex	attributes
1	A, C
2	A, C
3	A, C, D
4	A, D
5	A, C, E
6	A, B, C
7	A, B, E
8	A, B
9	A, B
10	A, B, D
11	A, B, C



[Silva et al., VLDB'12]

Attribute-Structure Correlated Patterns

- ❑ Definition
  - o A dense subgraph induced by a particular attribute set.

- ❑Characteristics
  - o High correlation between a given attribute set and the occurrence of dense subgraphs

$$\epsilon(S) = \frac{|\mathcal{K}_S|}{|\mathcal{V}(S)|}$$

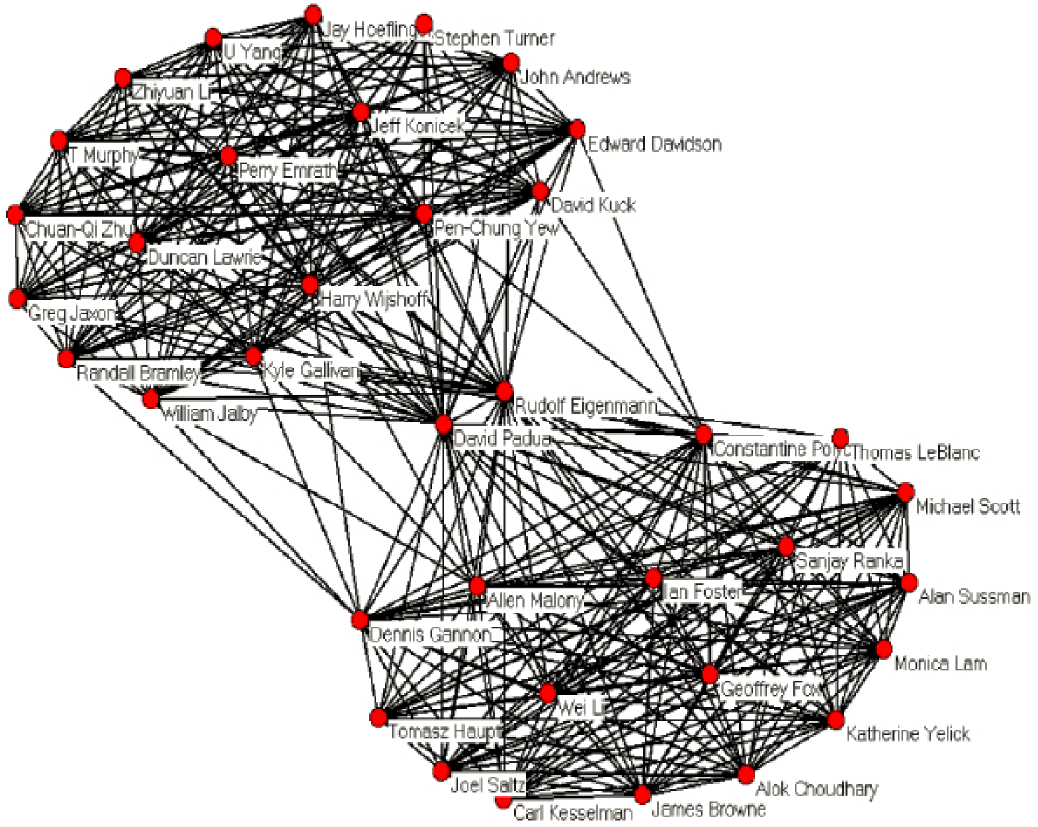
S: attribute set;
Ks: the set of vertices having *S* in dense subgraphs
V(*s*): the vectices with attribute set *S*

  - o High-support attribute sets do not necessarily present high structural correlation.

[Silva et al., VLDB'12]

Attribute-Structure Correlated Patterns

- DBLP
  - o 108,030 vertices
  - o 276,658 edges
  - o 23,285 attributes

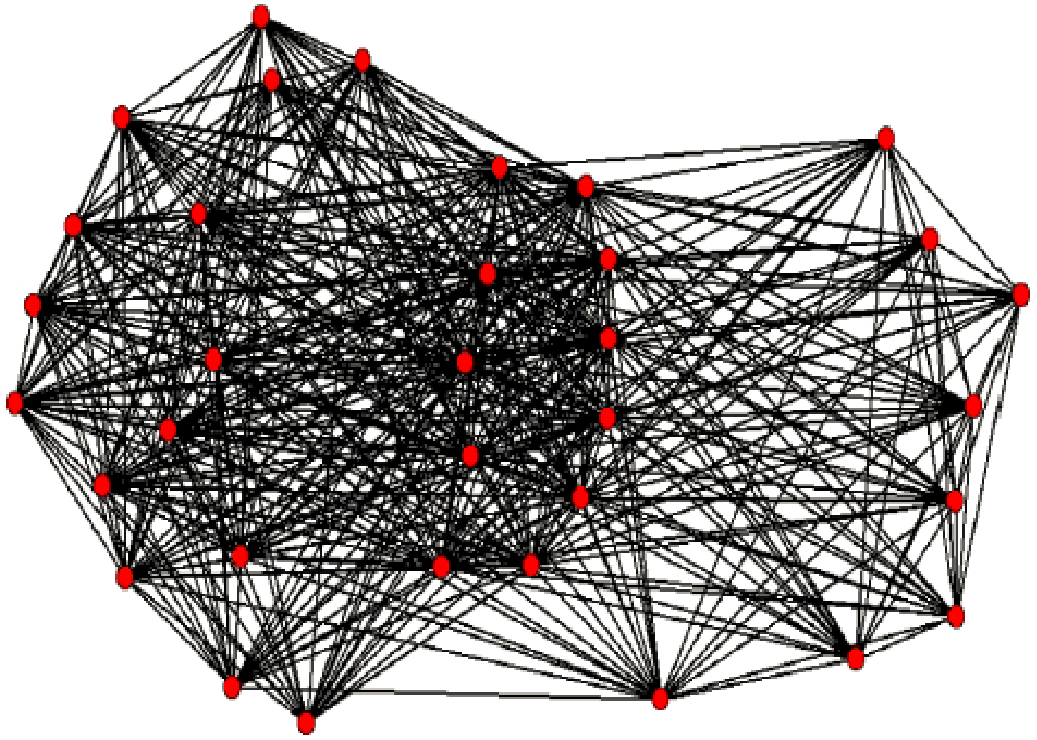


Pattern induced by {system , performance}

[Silva et al., VLDB’12]

Attribute-Structure Correlated Patterns

- Last.Fm
  - 272,412 vertices
  - 350,239 edges
  - 3,929,101 attributes

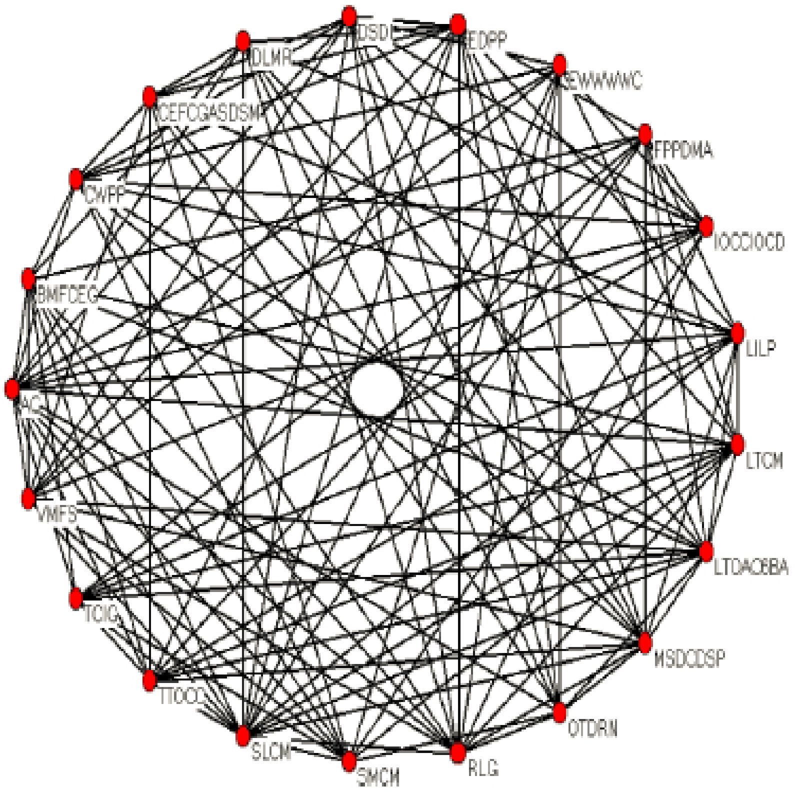


Pattern induced by {Van Morrison}

[Silva et al., VLDB’12]

Attribute-Structure Correlated Patterns

- ❑ CiteSeer
  - o 294,104 vertices
  - o 782,147 edges
  - o 206,430 attributes



Pattern induced by {perform, system}

[Guan et al., VLDB’11]

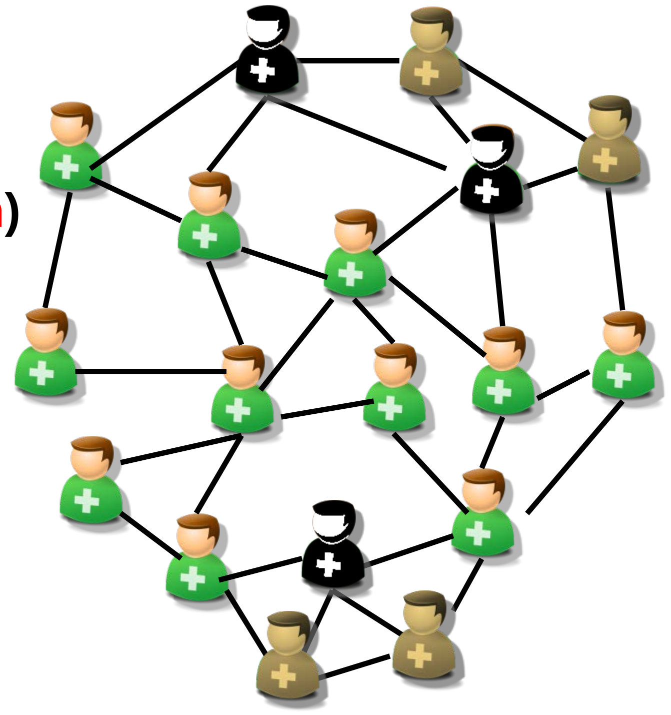
Two-event Structural Correlations

[Guan et al., VLDB’11]

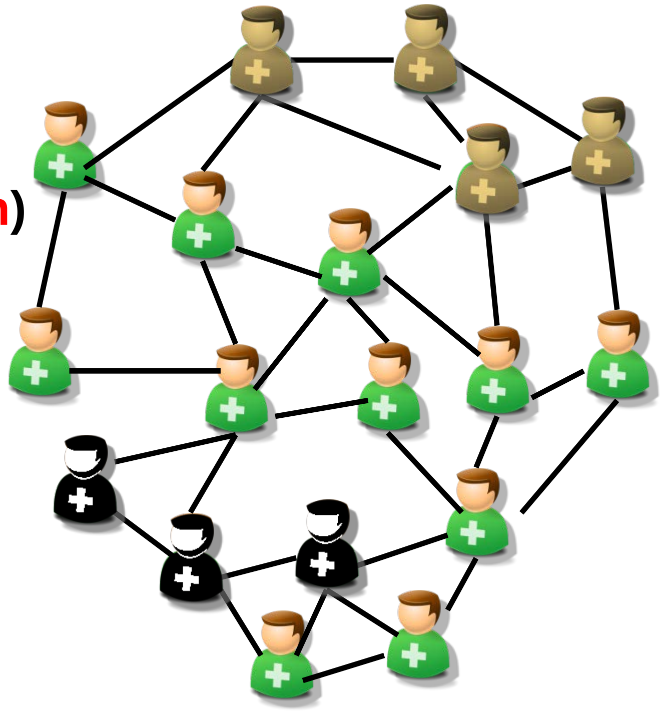
Two-event Structural Correlations

How is the relationship between the sales of two products in a social network?

Attraction (positive correlation)



Repulsion (negative correlation)



Video games

Computers

[Guan et al., VLDB'11]

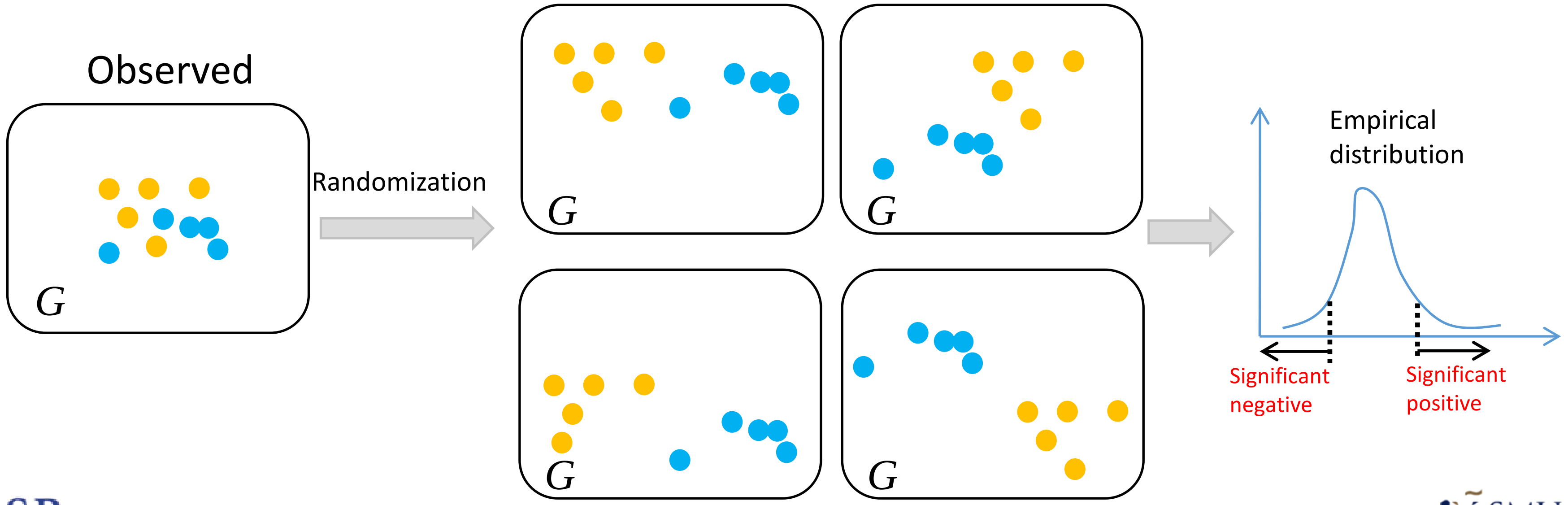
A New Notion of Correlation

- Two-Event Structural Correlation (**TESC**)
  - Defined on graph structures
  - Capture relationships between distributions of two events on a graph
  - Events can be different things in different contexts:
    - Topics or products (social networks)
    - Virus (contact networks)
    - Intrusion alerts (computer networks)

[Guan et al., VLDB’11]

The Major Challenge

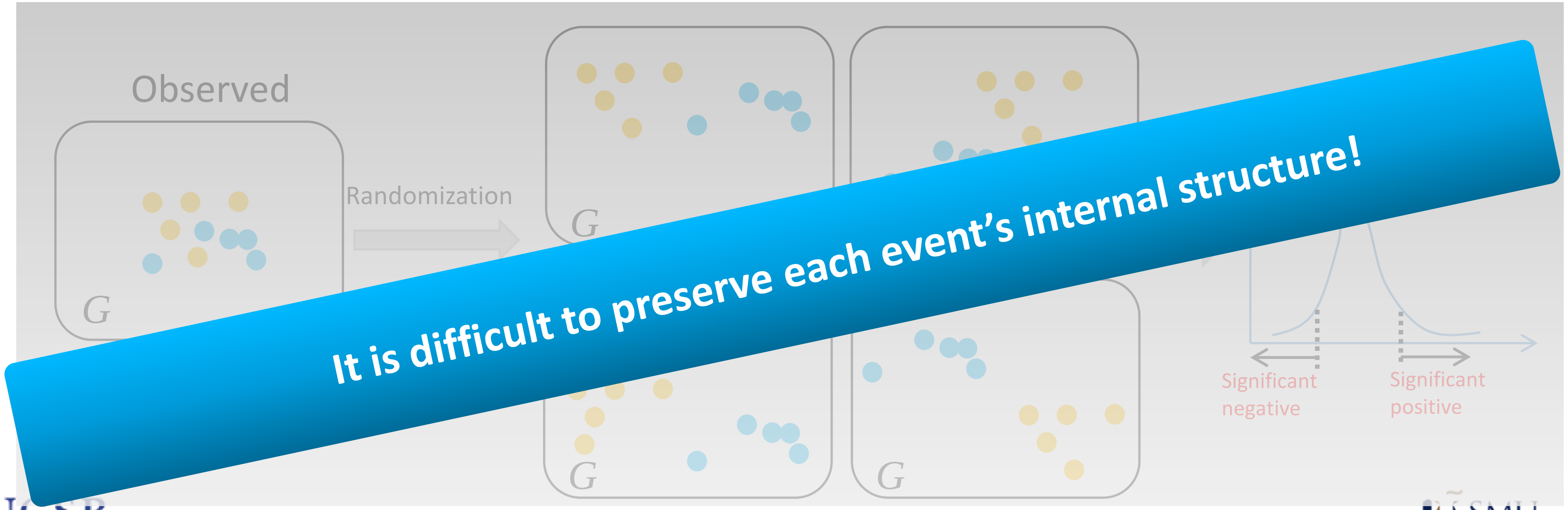
- ❑ **Simple idea**: compute the distance between the two events on the graph
  - o How near/far for **significant** positive/negative correlation?
  - o Hypothesis testing



[Guan et al., VLDB'11]

The Major Challenge

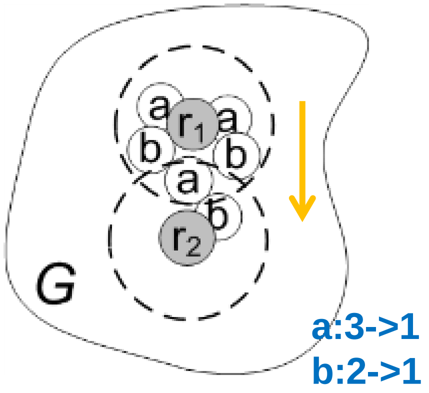
- **Simple idea**: compute the distance between the two events on the graph
  - How near/far for **significant** positive/negative correlation?
  - Hypothesis testing

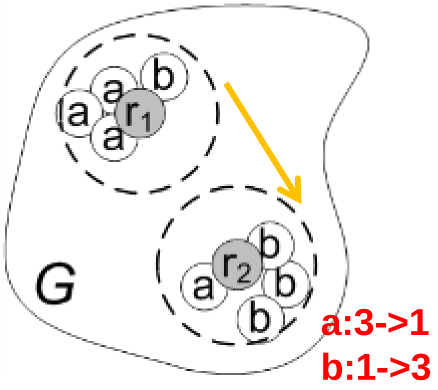


How To Measure?

"reference nodes"

🟦 **Consistent** 🟥 **Inconsistent**





- Concordance score

$$c(r_i,r_j)=\begin{cases}1 & (s_a^h(r_i)-s_a^h(r_j))(s_b^h(r_i)-s_b^h(r_j))>0\\ -1 & (s_a^h(r_i)-s_a^h(r_j))(s_b^h(r_i)-s_b^h(r_j))<0\\ 0 & otherwise\end{cases}$$

If the density changes are consistent

If the density changes are inconsistent

- Density function $s_a^h(r)=\frac{|V_a \cap V_r^h|}{|V_r^h|}$

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

[Guan et al., VLDB'11]

Kendall's Tau as The Measure

- **Kendall's Tau rank correlation** as a measure of TESC:

$$\tau(a,b) = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c(r_i, r_j)}{\frac{1}{2} N(N-1)}$$

$N = |V^h_{a \cup b}|$: the number of all reference nodes

- Costly computation!

Sampling + significance testing

Real Events (DBLP)

Highly positive pairs:

#	Pair	z-score			TC
		h = 1	h = 2	h = 3	
1	Texture **vs.** Image	6.22	19.85	30.58	172.7
2	Wireless **vs.** Sensor	5.99	23.09	32.12	463.7
3	Multicast **vs.** Network	4.21	18.37	26.66	123.2
4	Wireless **vs.** Network	2.06	17.41	27.90	198.2
5	Semantic **vs.** RDF	1.72	16.02	24.94	120.3

Highly negative pairs:

#	Pair	z-score			TC
		h = 1	h = 2	h = 3	
1	Texture **vs.** Java	-23.63	-9.41	-6.40	4.33
2	GPU **vs.** RDF	-24.47	-14.64	-6.31	1.24
3	SQL **vs.** Calibration	-21.29	-12.70	-5.45	-0.62
4	Hardware **vs.** Ontology	-22.31	-8.85	-5.01	3.38
5	Transaction **vs.** Camera	-22.20	-7.91	-4.26	4.85

z-score > 2.33: significant positive TESC
z-score < -2.33: significant negative TESC

TC: Transaction Correlation

[Guan et al., VLDB’11]

Several Other Pattern Miners

- ❑ CoPaM: Cohesive Pattern Miner [Moser et al., SDM’09]
- ❑ Itemset-Sharing Patterns [Fukuzaki et al., 2013]
- ❑ Topological Pattern Miner [Prado et al., TKDE’13]

[Moser et al., SDM'09]

CoPaM: Cohesive Pattern Miner

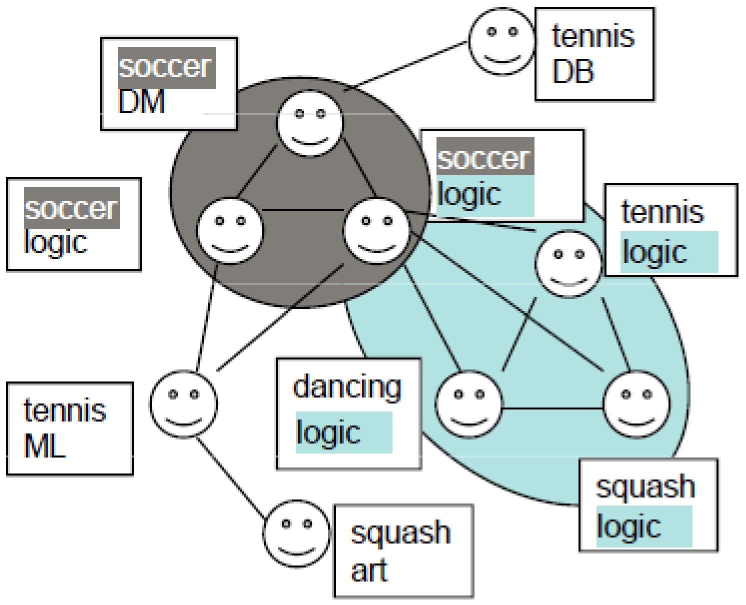
- Cohesive Pattern

  A connected subgraph satisfying:

  (1) Feature subspace cohesion

  (2) Densely connected

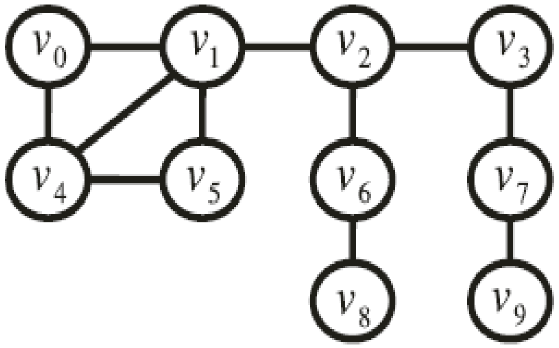
Two cohesive patterns:



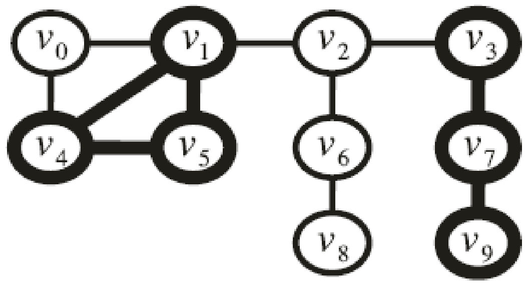
[Fukuzaki et al., PAKDD’10]

Itemset-Sharing Patterns

- Goal
  - To find sets of subgraphs with common itemsets in a large graph.



v_0	$\{i_1,i_3\}$	v_5	$\{i_1,i_2,i_3,i_4\}$
v_1	$\{i_1,i_2,i_4\}$	v_6	$\{i_2,i_4,i_5\}$
v_2	$\{i_2,i_5\}$	v_7	$\{i_1,i_2,i_3\}$
v_3	$\{i_1,i_2,i_3\}$	v_8	$\{i_1,i_2\}$
v_4	$\{i_1,i_2,i_4\}$	v_9	$\{i_1,i_2,i_3,i_4\}$



(a)
Itemset –attributed graph

(b)
Itemset-sharing subgraphs with $\{i_1,i_2\}$.

[Prado et al., TKDE'13]

Topological Pattern Miner

- A Topological Pattern

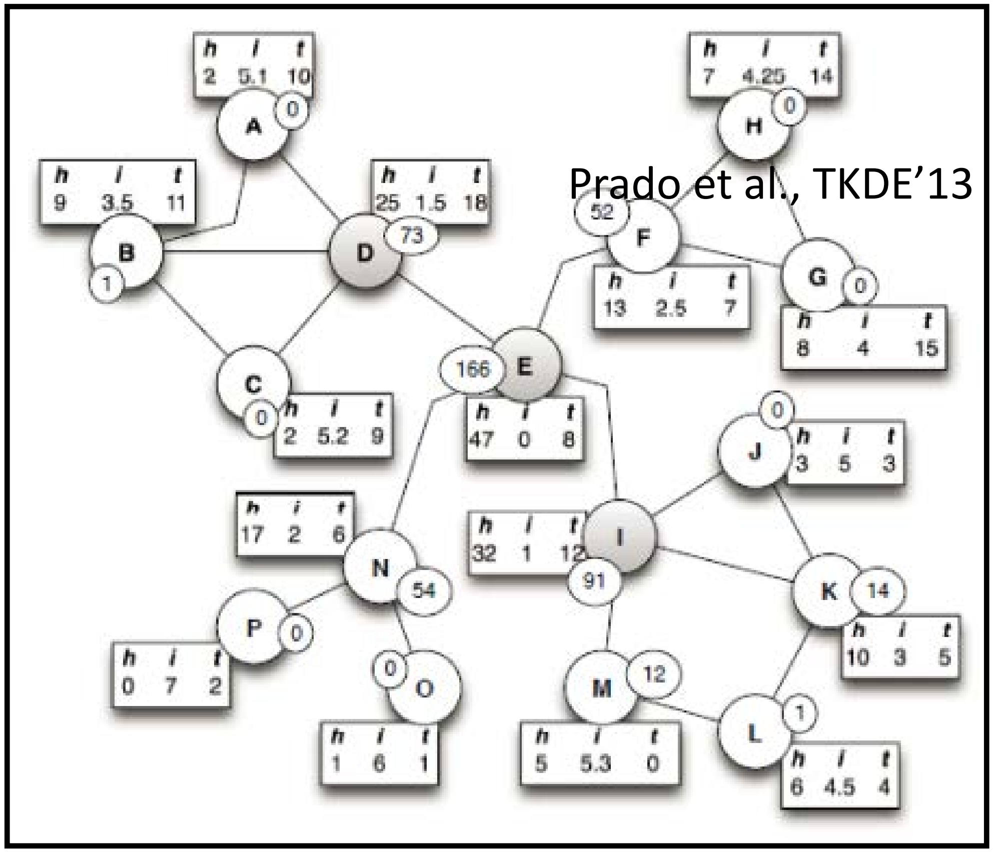
A set of vertex attributes and topological properties that strongly co-vary over the vertices of the graph

[Prado et al., TKDE’13]

Topological Pattern Miner

- A Topological Pattern

A set of vertex attributes and topological properties that strongly co-vary over the vertices of the graph



$$P = \{h^+, i^-, \textsc{Betw}^+\}$$

The higher the value of attribute *h*, the lower the value of attribute *i* and the higher the betweenness centrality of a vertex.

Anomaly Detection in Graphs

- Various Interesting-ness/Anomaly Criteria

  e.g.,

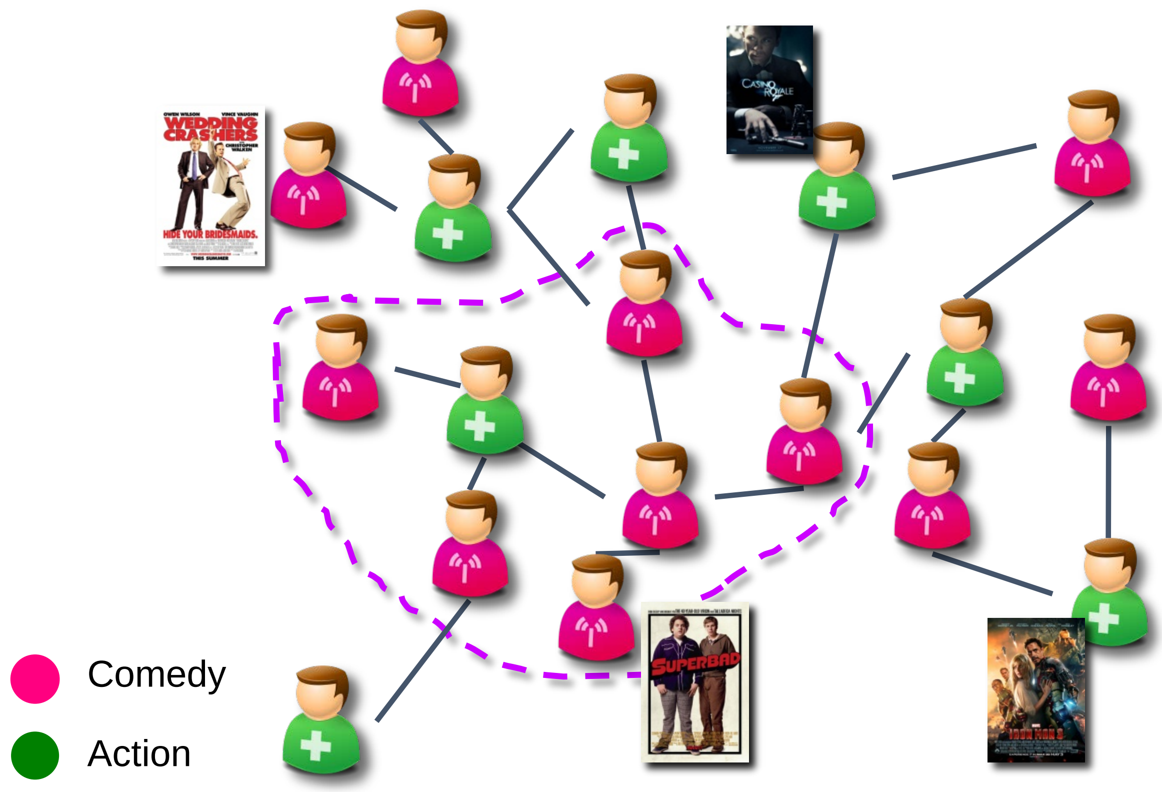
  o Bgp-lens: anomalies in internet routing updates.
  [Prakash et al., KDD'09]

  o Oddball: anomalies in weighted graphs.
  [Akoglu et al., PAKDD'10]

  o Heavy subgraphs in time-evolving networks.
  [Bogdanov et al., ICDM'11]

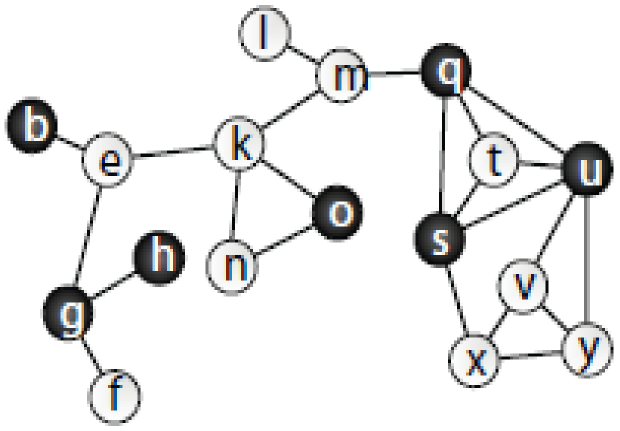
- Anomaly, Event, and Fraud Detection in Large Graph Datasets, Akoglu et al., http://www.cs.stonybrook.edu/~leman/wsdm13/

Anomaly Vertices/ Regions

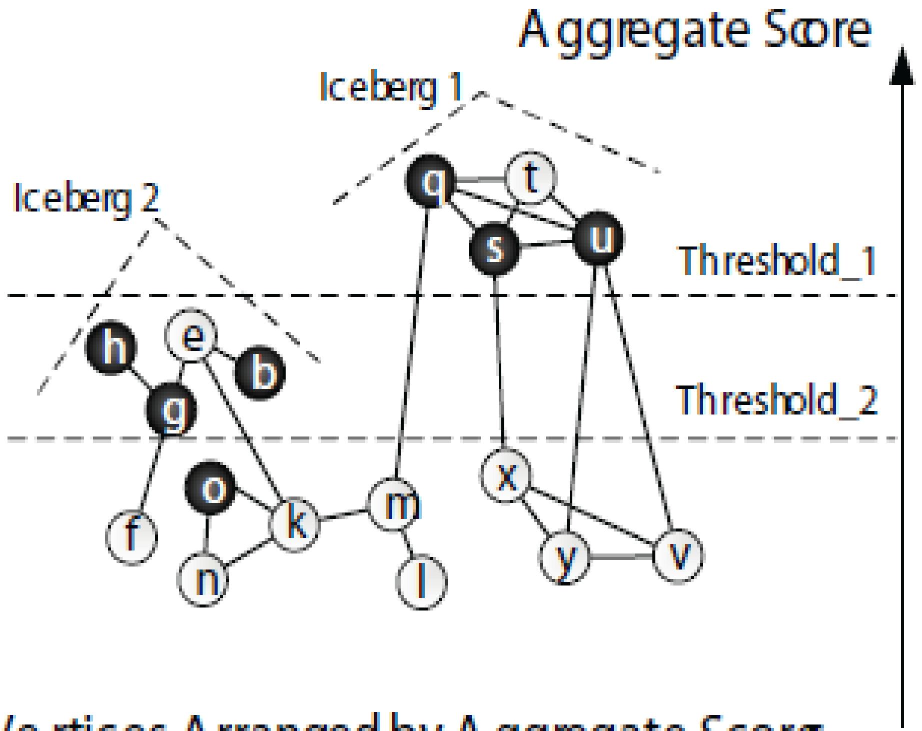


1. Target marketing
2. Recommendation systems
3. Social influence analysis

Graph Iceberg (i.e., gIceberg)



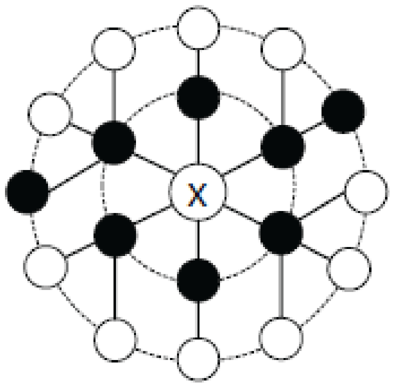
(a) Original Graph



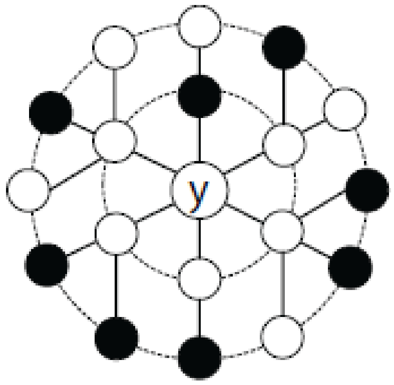
(b) Vertices Arranged by Aggregate Scores

What aggregate functions?

- SUM and AVG [Yan et al., ICDE'10]



(a) x's 2-hop neighborhood

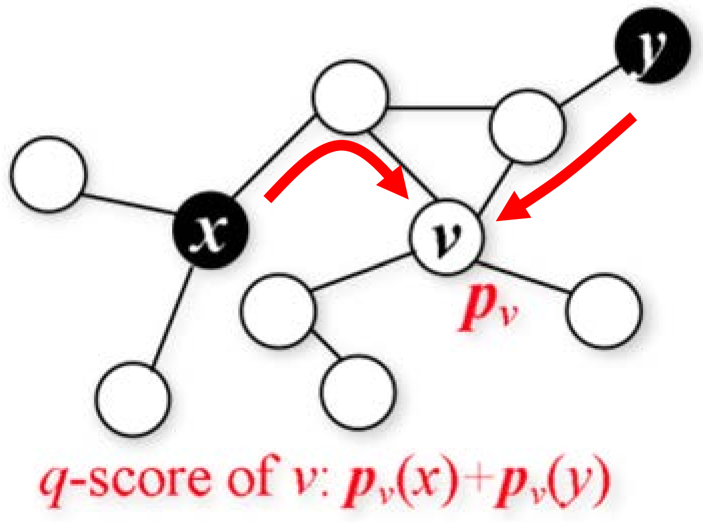


(b) y's 2-hop neighborhoo d

- Personalized PageRank Vector (PPV) [Page et al., Technical Report, 1999]
  - Reflect the proximity from one node to another w.r.t. the graph structure.

[Li et al., ICDE'13]

Aggregate q-score:



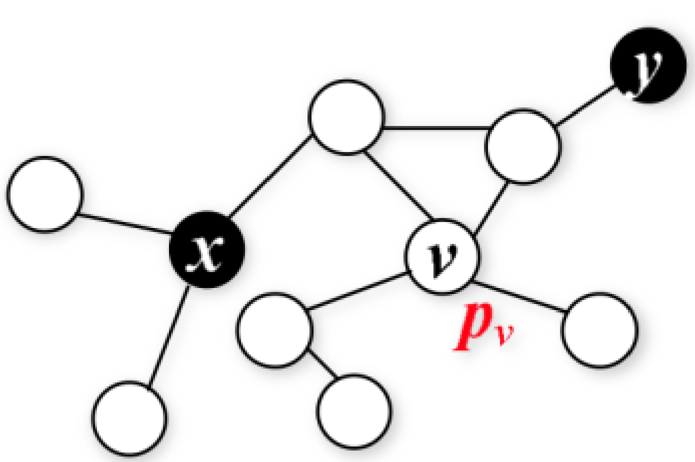
q-score of v: $p_v(x)+p_v(y)$

$$\boxed{\mathcal{P}_q(v) = \Sigma_{x|x\in V, q\in L(x)} p_v(x)}$$

Aggregate over the Personalized PageRank Vector (PPV) to define the *closeness* of a vertex and an attribute *q.*

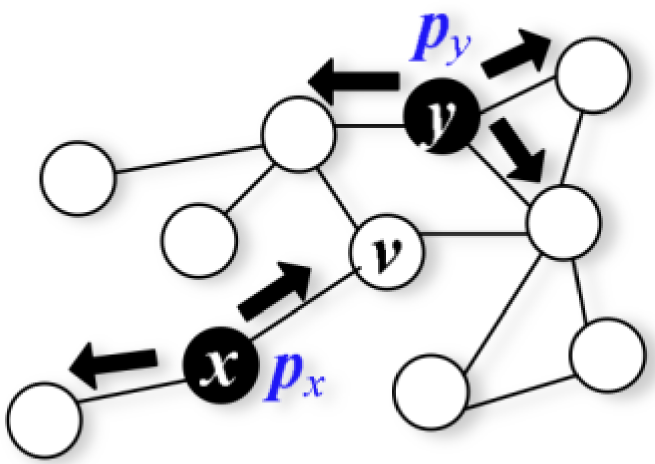
[Li et al., ICDE'13]

Forward & Backward Aggregation



q-score of v: $p_v(x)+p_v(y)$

(*a*) **FA**

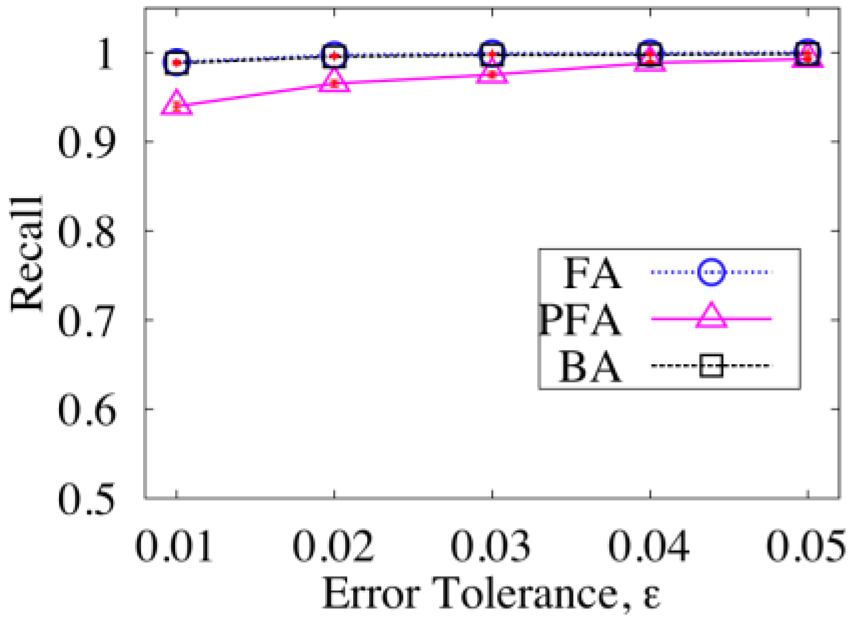


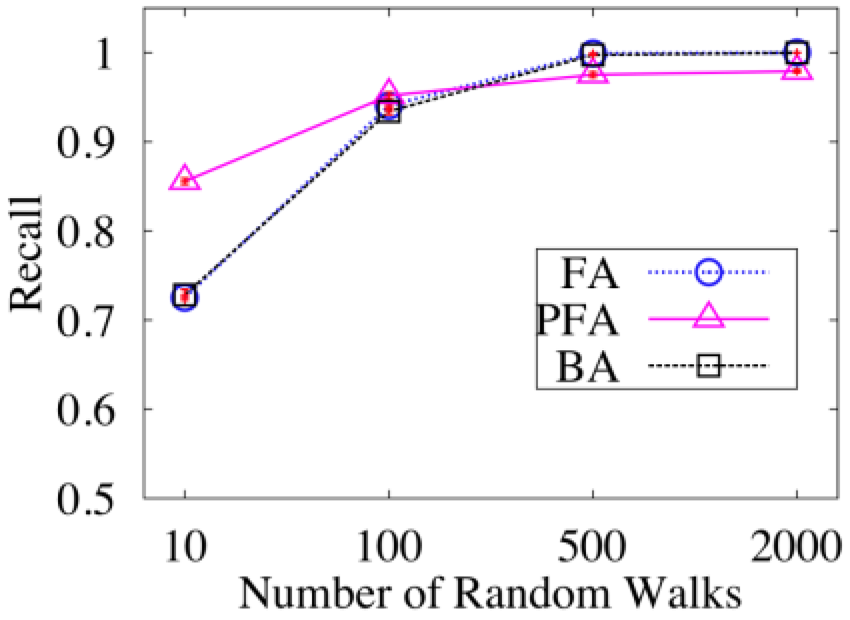
q-score of v: $f_x(p_x(v))+f_y(p_y(v))$

(*b*) **BA**

Results in Customer Network

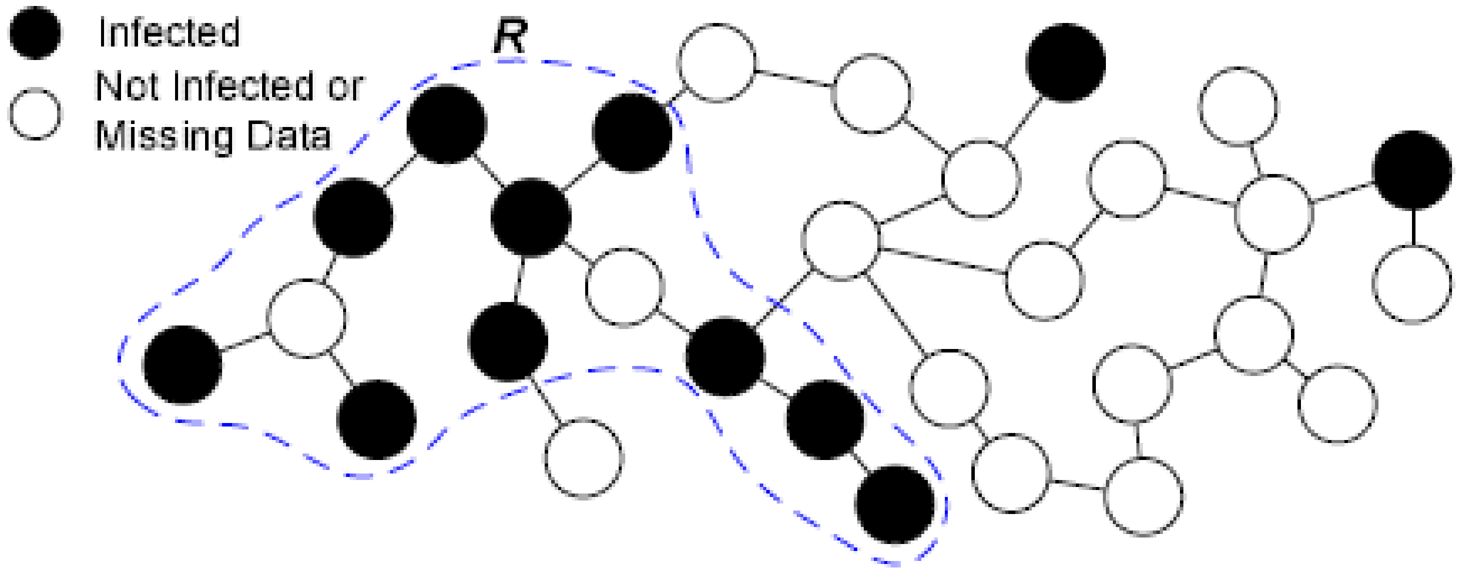
- ❑ Recall indicates gIceberg's ability to retrieve real graph Iceberg vertices.





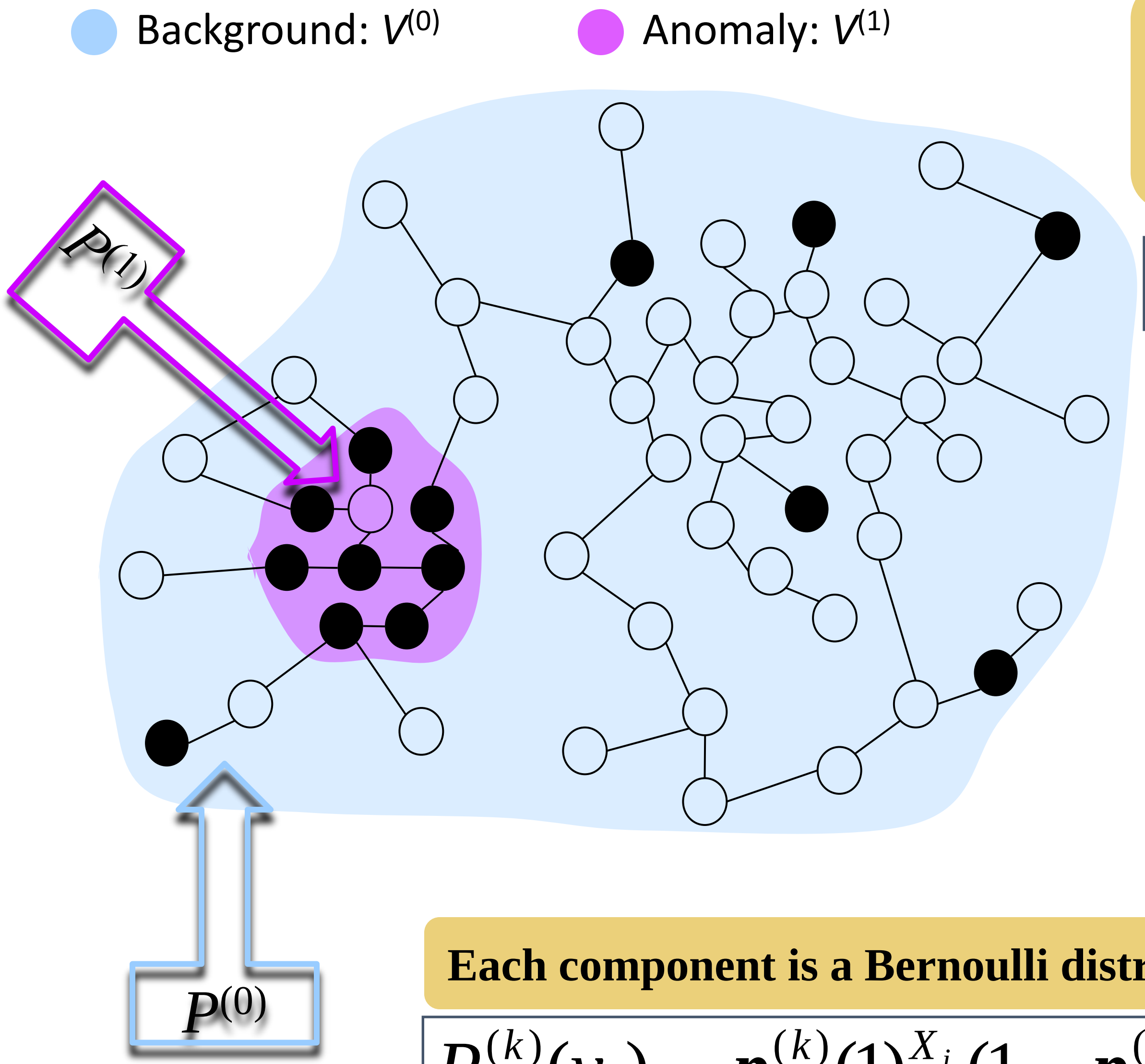
[Li et al., SDM'14]

Anomalous Regions (i.e., gAnomaly)



- ❑ Why does a disease occur more intensively in some portions of a network?
- ❑ Why do a subset of computers receive most of the attacks in the past day, and are they therefore targeted attacks?

Data Models



Two generative processes: anomaly distribution & background distribution

$$V = V^{(0)} \bigcup V^{(1)}, V^{(0)} \bigcap V^{(1)} = \varnothing$$

One overall mixture

$$P(v_i) = \sum_{k=0}^{1} \theta_i^{(k)} P^{(k)}(v_i)$$

With probability $\vartheta_i^{(0)}$, v_i belongs to the background component $V^{(0)}$, and with $\vartheta_i^{(1)}$ the anomaly component $V^{(1)}$.

Each component is a Bernoulli distribution.

$$P^{(k)}(v_i) = \boldsymbol{p}^{(k)}(1)^{X_i} (1 - \boldsymbol{p}^{(k)}(1))^{1-X_i}$$

UCSB SMU SINGAPORE MANAGEMENT UNIVERSITY

[Li et al., SDM'14]

Regularized Data Likelihood

- Un-regularized log data likelihood of vertex set *V*

$$L(V) = \sum_{v_i \in V} \log P(v_i) = \sum_{v_i \in V} \log \sum_k \theta_i^{(k)} P^{(k)}(v_i)$$

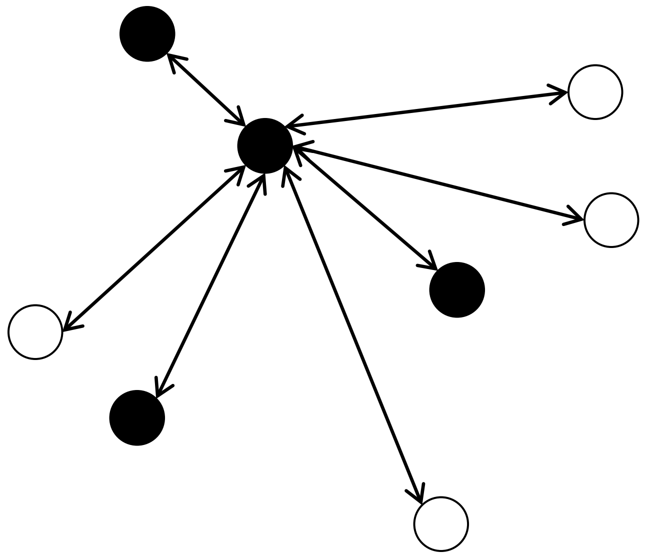
- Regularized log data likelihood of vertex set *V*

$$\widehat{L}(V) = L(V) - \lambda R_N(\Theta) + \gamma R_E(\Theta)$$

  - Enhance connectivity within each component
    - Network regularizer
  - Enhance the interpretability of the mixture weights
    - Entropy regularizer

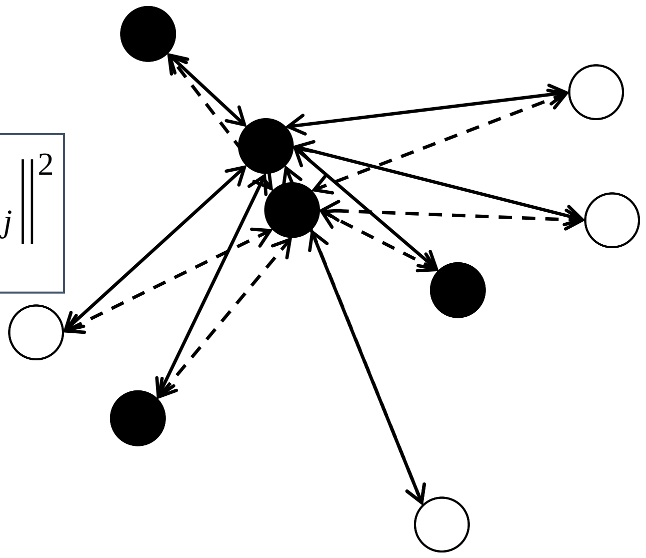
[Li et al., SDM'14]

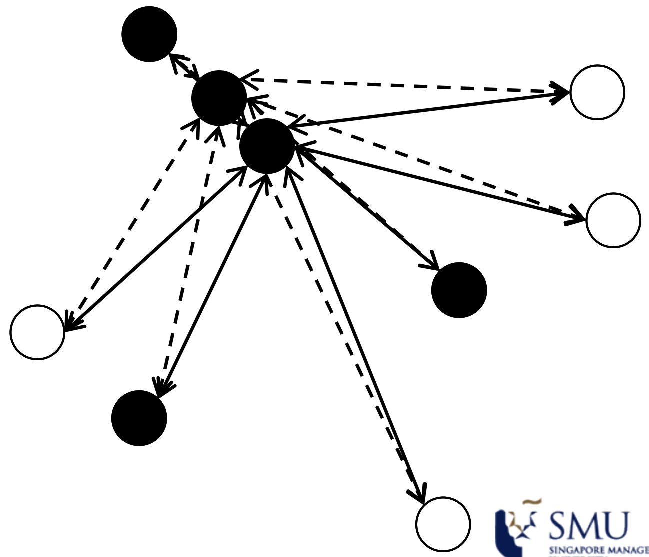
Network Regularizers



Minimize the mean

$$R_N^{(1)}(\Theta) = \frac{1}{2}\sum_{v_i \in V}\frac{1}{|N(i)|}\sum_{v_j \in N(i)}\left\|\Theta_i - \Theta_j\right\|^2$$





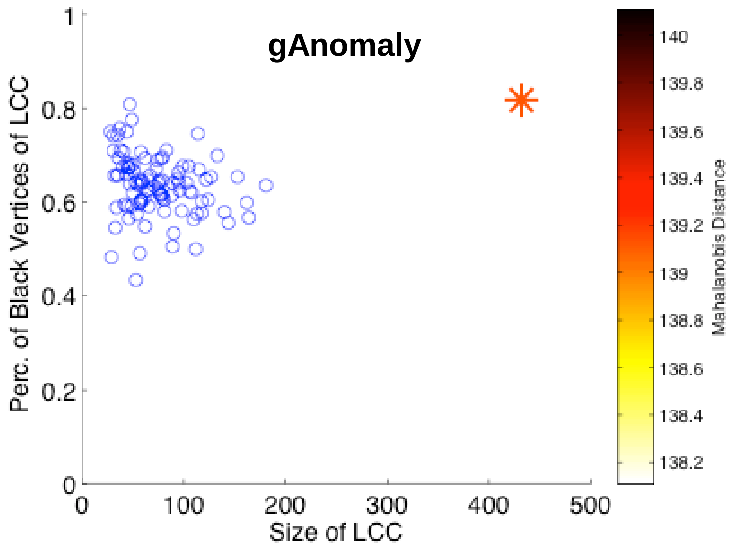
Minimize the minimum

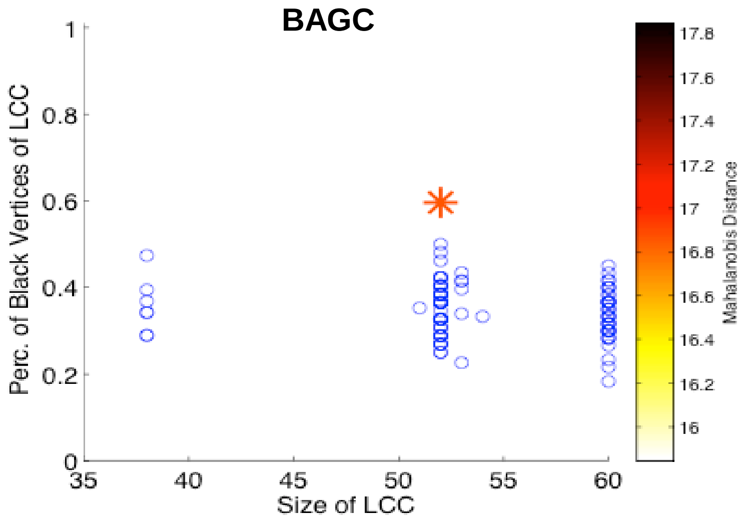
$$R_N^{(2)}(\Theta) = \frac{1}{2}\sum_{v_i \in V}\min_{v_j \in N(i)}\left\|\Theta_i - \Theta_j\right\|^2$$

[Li et al., SDM'14]

Significance Evaluation

M-distance Comparison



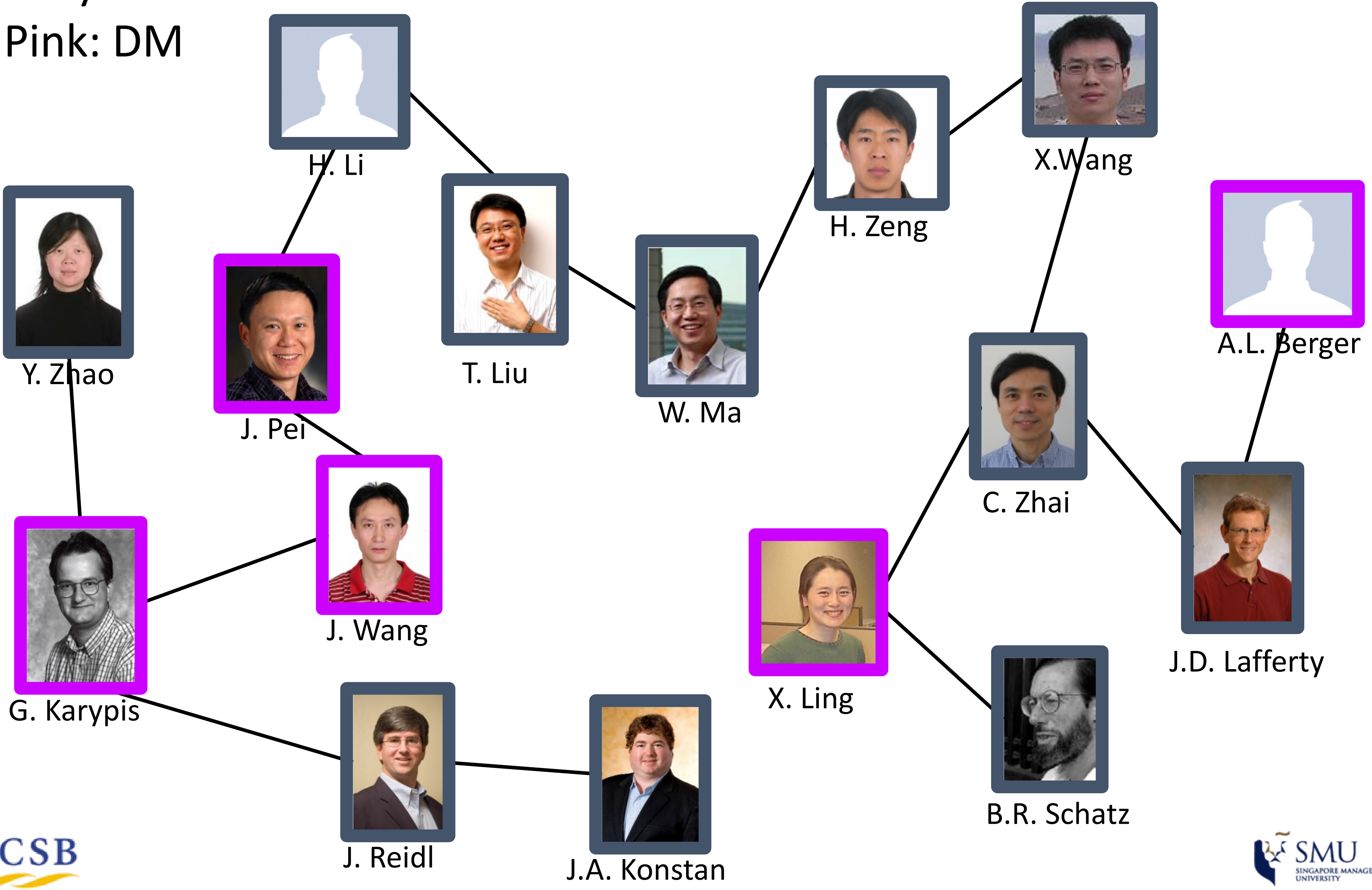


[Li et al., SDM'14]

Case Study on DBLP-IR

Grey: IR

Pink: DM



Part I: Brief Sum-Up

- ❑ Proximity Pattern ✓
- ❑ Attribute-Structure Correlations ✓
- ❑ Cohesive Pattern ✓
- ❑ Itemset-sharing Pattern ✓
- ❑ Graph Topological pattern ✓
- ❑ Graph Anomaly ✓
- ❑ Frequent Network Pattern (Part II)

References

- Towards Proximity Pattern Mining in Large Graphs. [Khan et al., SIGMOD'10]
- Assessing and ranking structural correlations in graphs. [Guan et al., SIGMOD'11]
- Measuring Two-Event Structural Correlations on Graphs. [Guan et al., VLDB'11]
- Mining Attribute-structure Correlated Patterns in Large Attributed Graphs. [Silva et al., VLDB'12]
- Mining Cohesive Patterns from Graphs with Feature Vectors. [Moser et al., SDM'09]
- Finding Itemset-Sharing Patterns in a Large Itemset-Associated Graph. [Fukuzaki et al., PAKDD'10]

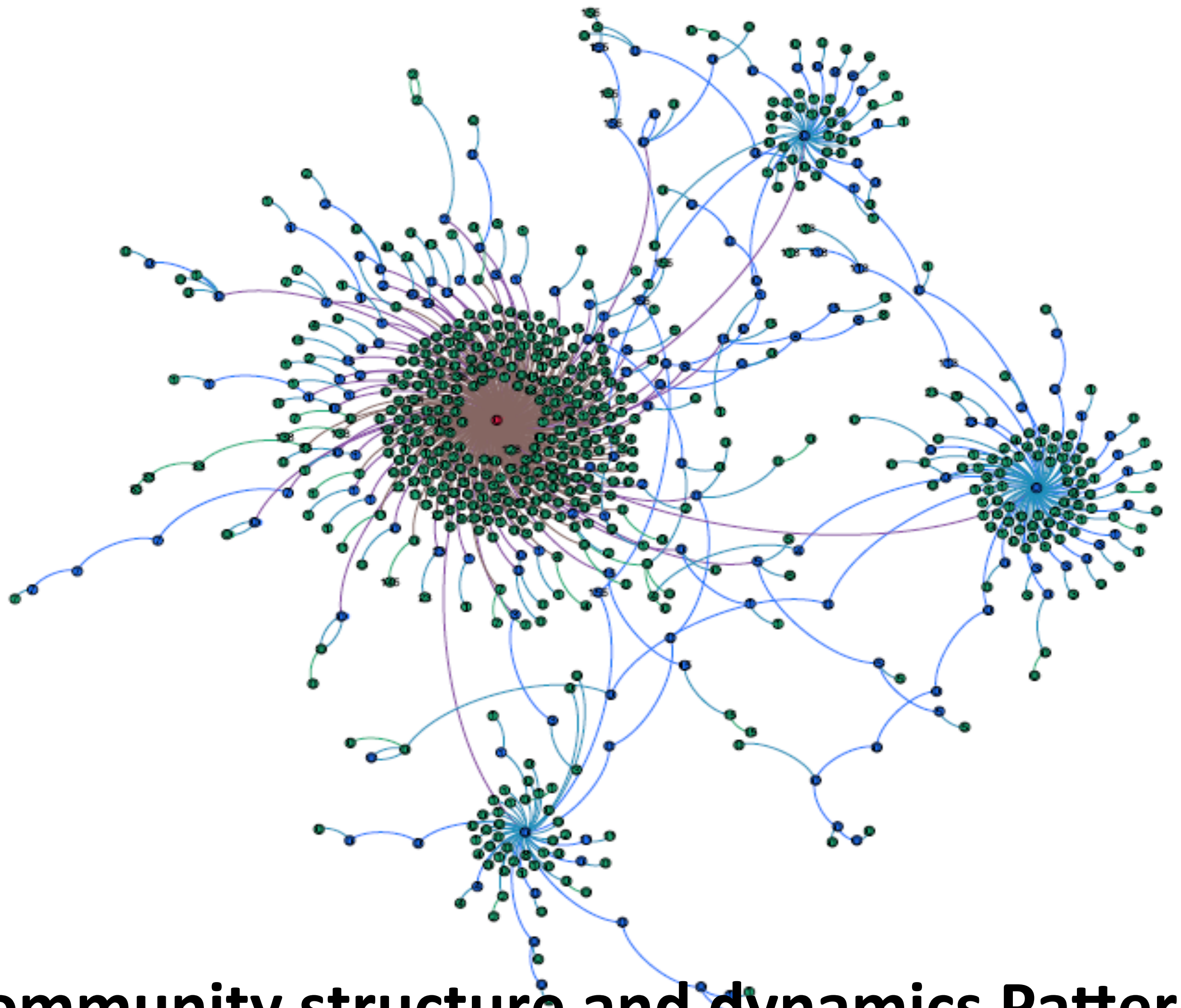
References

- Mining graph topological patterns: Finding covariations among vertex descriptors. [Prado et al., TKDE’13]
- Bgp-lens: anomalies in internet routing updates. [Prakash et al., KDD’09]
- Oddball: Anomalies in Weighted Graphs. [Akoglu et al., PAKDD’10]
- Heavy Subgraphs in Time-Evolving Networks. [Bogdanov et al., ICDM’11]
- Giceberg: Towards Iceberg Analysis in Large Graphs. [Li et al., ICDE’13]
- A Probabilistic Approach to Uncovering Attributed Graph Anomalies. [Li et al., SDM’14]

Part II

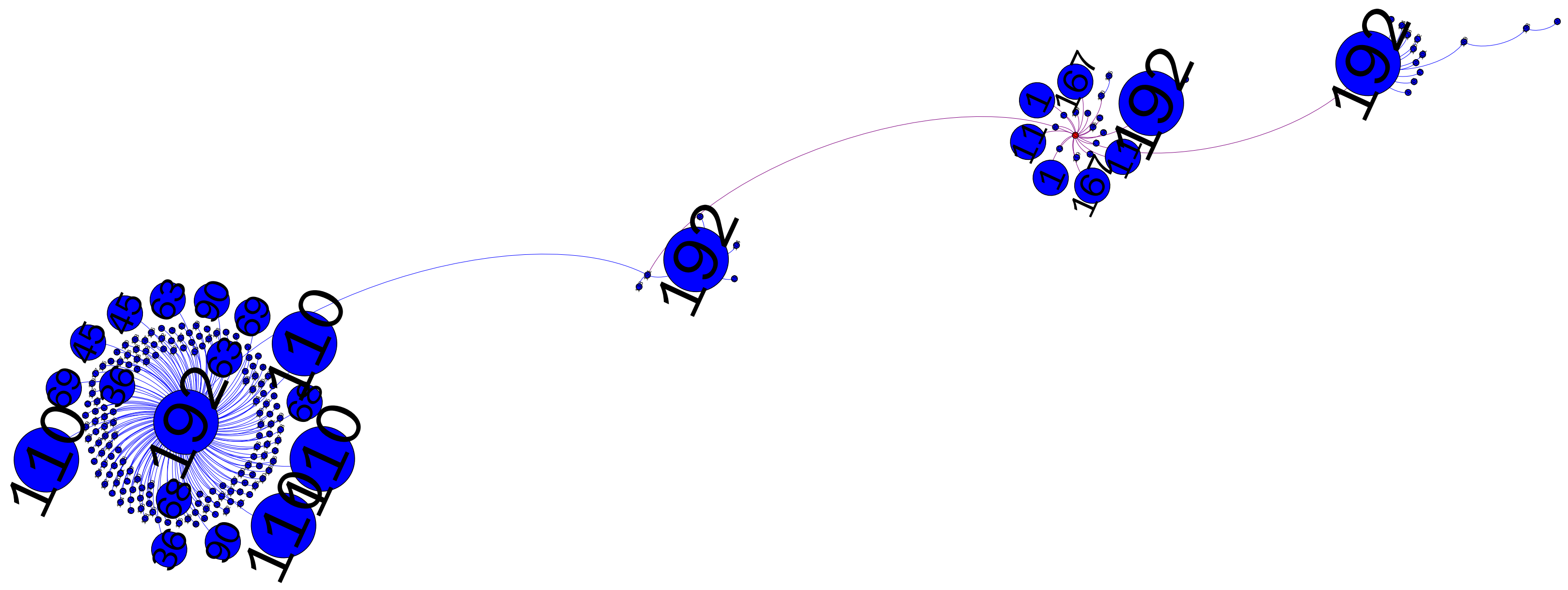
Frequent Network Pattern for Social Applications

Frequent Network Patterns for Social Applications



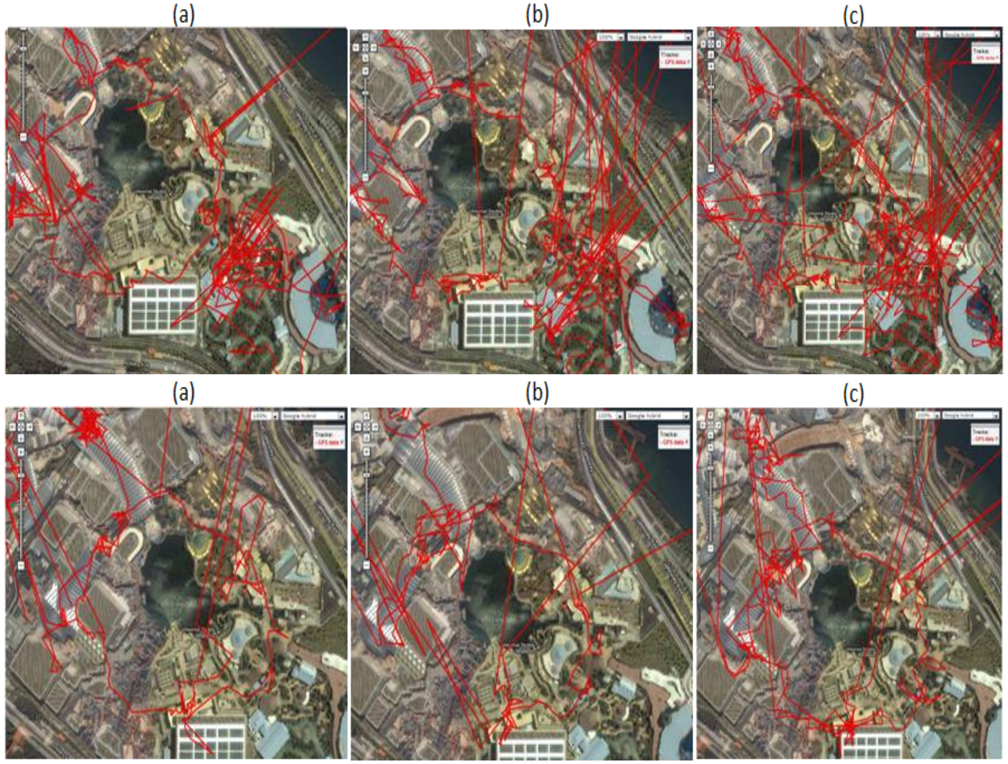
Community structure and dynamics Patterns

Frequent Network Patterns for Social Applications



Information Diffusion and Viral Marketing Patterns

Frequent Network Patterns for Social Applications



Mobile User Trajectory and Location Patterns

Some Important Characteristics

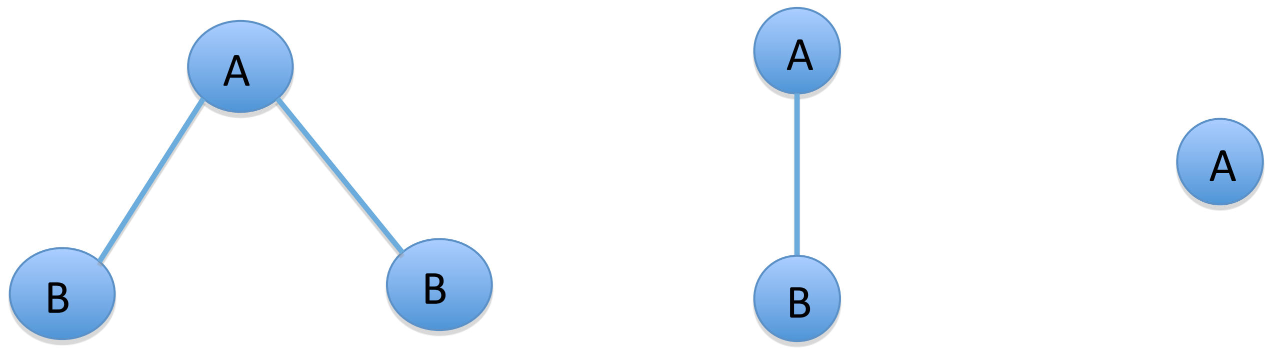
- ❑ Single-graph setting

- ❑ Huge network size

- ❑ User-specified constraints

Single network mining is not easy

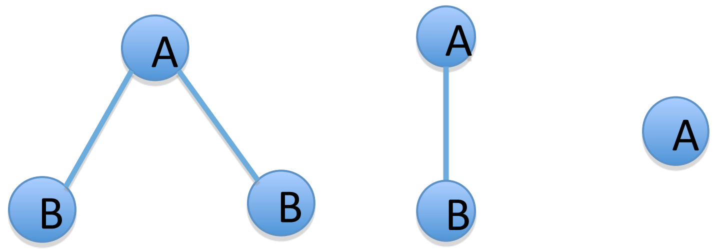
- Support counting is tricky when more than one pattern embedding would be considered in a single data graph.

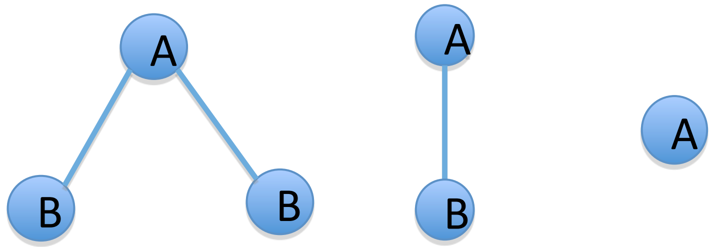


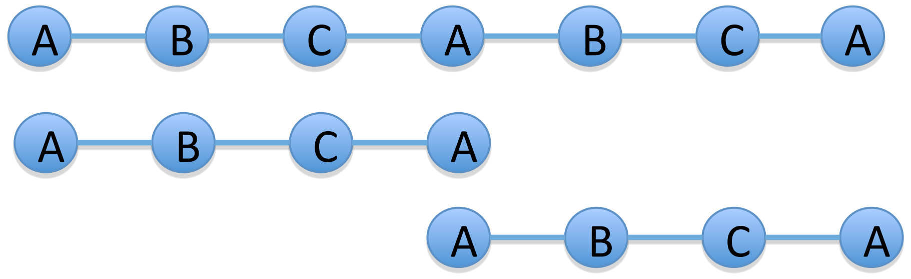
- The anti-monotone property of support is violated.
- All the frequency-based pruning techniques must fail.
- No infrequent subgraph patterns can be pruned.

How to fix it?

- Edge-disjoint
- Node-disjoint
  - Overlap graph
  - Maximum Independent Set
  - MIS-support
    - Kuramochi and Karipus
  - MIS-support is antimonotone
  - But MIS-support is too restrictive!
- Harmful overlap support
  - Fiedler and Borgelt





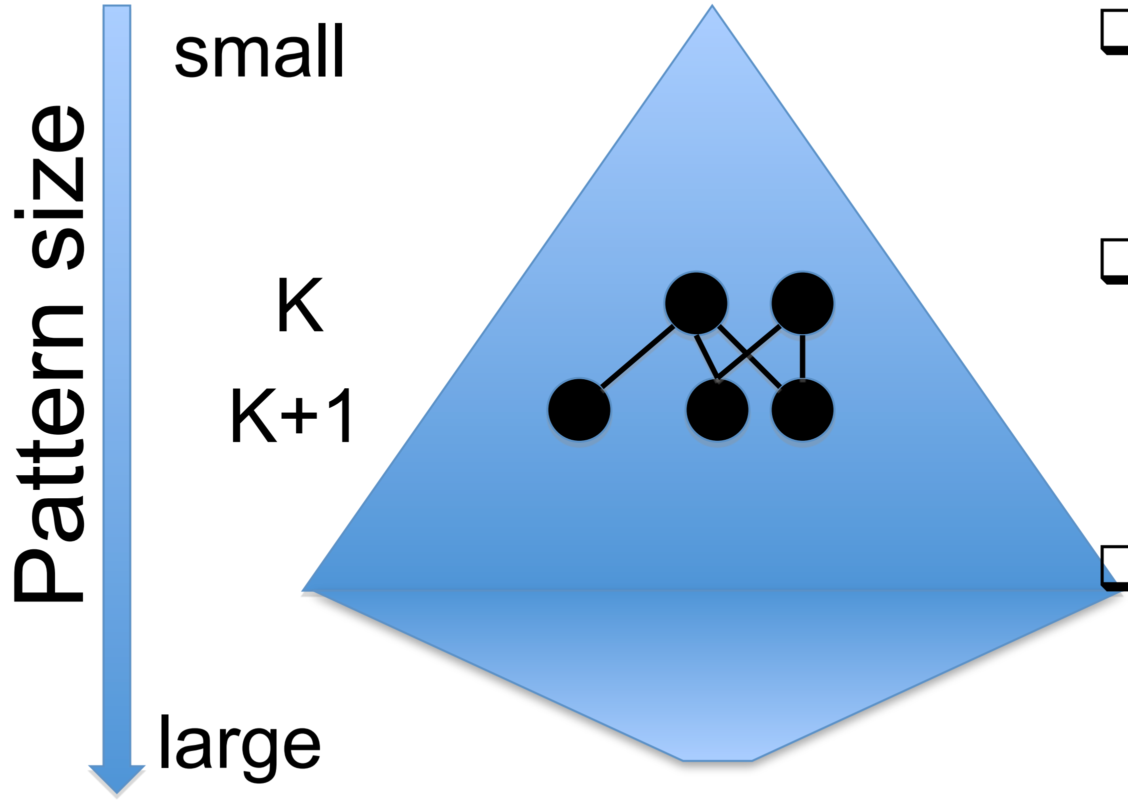


Some Important Characteristics

- ❑ Single-graph setting
- ❑ Huge network size
- ❑ User-specified constraints

Not all patterns are wanted

A Pattern Lattice Perspective



- Smaller patterns are affordable to be enumerated
- The number of patterns quickly explodes as pattern size grows larger
- The number of large patterns is small

It's all about different ways to traverse this pattern lattice.

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

Some Important Characteristics

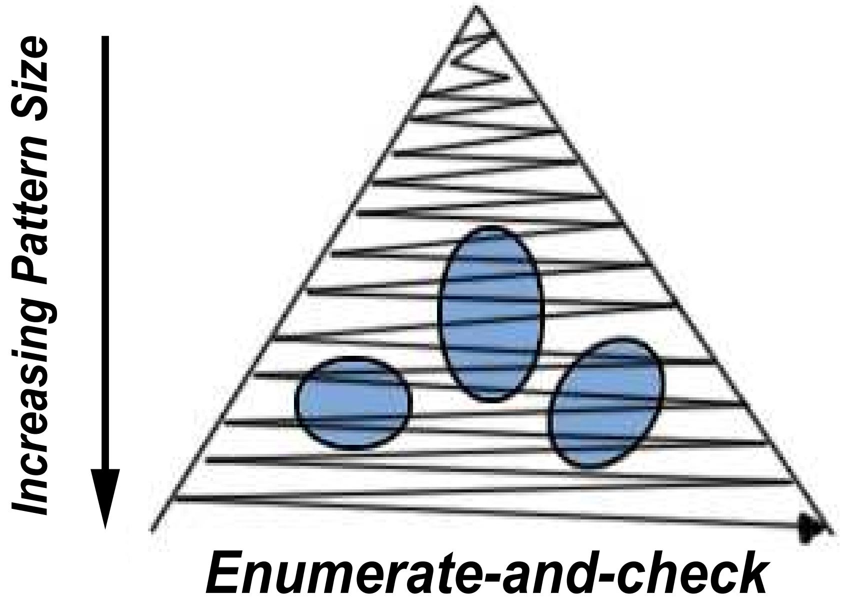
- ❑ Single-graph setting

- ❑ Huge network size

- ❑ User-specified constraints

Traditional Solution for User-specified Constraints

- ❑ Mine all frequent patterns
  - ❑ Pattern candidate generation
  - ❑ Frequency checking in the data graph
- ❑ Check each frequent pattern for constraints
- ❑ Output those constraint-satisfying ones



Problem: pattern explosion!

What we really need ---- Blended frequent pattern mining with constraint-checking

Requirement Summary

- Single-graph setting
  - Be able to handle both single-graph setting and graph-transaction setting.
- Huge network size
  - Avoid enumerating all the pattern candidates before reaching those of interest.
- User-specified constraints
  - The mining should be "direct".

Representative Work

- ❑Graph-transaction setting
    - ❑AGM, FSG, gSpan, FFSM, etc.
        - ❑For mining *complete* pattern set.
        - ❑Suffers from scalability issue for large patterns and input graphs due to exponential result size.
    - ❑SPIN and MARGIN
        - ❑Mining *maximal* patterns.
        - ❑Still suffers from scalability issue as the number of maximal patterns could be formidable.
    - ❑ORIGAMI
        - ❑For mining a *representative* pattern set.
        - ❑Returns a pattern set of mixed sizes.

Representative Work

- Single-graph setting
  - SUBDUE and SEuS
    - Pioneer work for mining *complete* pattern set.
    - Use different heuristics and work well for mining smaller patterns on certain classes of input graphs.
  - GREW
    - Mining *incomplete* pattern set.
    - Able to discover some large patterns, yet no guarantee on the pattern coverage of the answer set.
  - MoSS
    - State-of-the-art for mining *complete* pattern set.
    - Suffers from scalability issue for large patterns and input graphs due to exponential result size.

SpiderMine: Frequent Large Patterns
[Zhu et al. PVLDB’11]

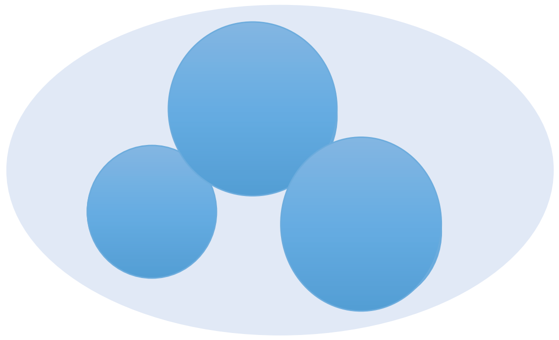
- Graph data is getting ever bigger, and so are the patterns.
    - E.g., social networks like Facebook, Twitter, etc..
- Often, large patterns are more informative in characterizing large graph data.
    - E.g., in DBLP, small patterns are ubiquitous, larger patterns better characterize different research communities.
    - E.g., in software engineering, large patterns reveal software *backbones*

Top-K largest frequent patterns

- To capture them exactly, no more and no less, we would have to generate all the smaller ones, which we cannot afford.
- A probabilistic solution, with user-defined error bound.
- SpiderMine --- designed to mine the top-K largest frequent patterns whose diameters are bounded by D_{max} with a probability at least $1-\varepsilon$

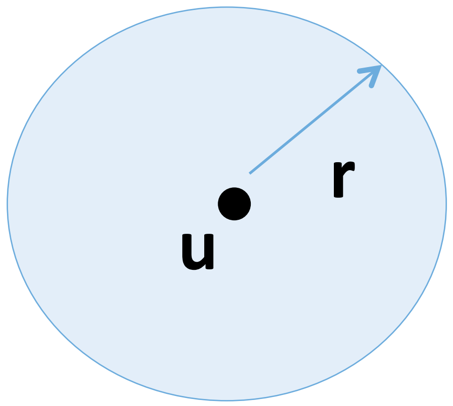
Design Ideas

- How to capture large graph patterns?
- Observation:
  - Large patterns are composed of a large number of small components, called “spiders”, which will eventually connect together after some rounds of pattern growth.



r-Spider

- An ***r-spider*** is a frequent graph pattern P such that there exists a vertex u of P, and all other vertices of P are within distance r to u. u is called the head vertex.



SpiderMine Overview

1) Mine the set S of all the r-spiders.
2) Randomly draw M r-spiders.
3) Grow these M r-spiders for t iterations. During the process, merge two patterns whenever possible.
4) Discard unmerged patterns.
5) Continue to grow the remaining ones to maximum size.
6) Return the top-K largest ones in the result.

◆ $t = D_{max}/2r$

Large patterns vs small patterns

- ❑ Why can SpiderMine preserve large patterns and prune small ones with good chance?

1) Small patterns are much less likely to be hit in the random draw.
   - ❖ First pruning at the initial random draw

2) Even if a small pattern is hit, it's even less likely to be hit multiple times.
   - ❖ Second pruning after t iteration growth

3) The larger the pattern, the greater the chance it is hit and saved.

How many r-spiders to draw?

LEMMA 2. *Given a network G and a user-specified K, we have* $P_{success} \geq \left(1 - (M+1)(1 - \frac{V_{min}}{|V(G)|})^M\right)^K$.

With user-defined error threshold ε, we solve for M by setting

$$\left(1 - (M+1)(1 - \frac{V_{min}}{|V(G)|})^M\right)^K = 1 - \epsilon$$

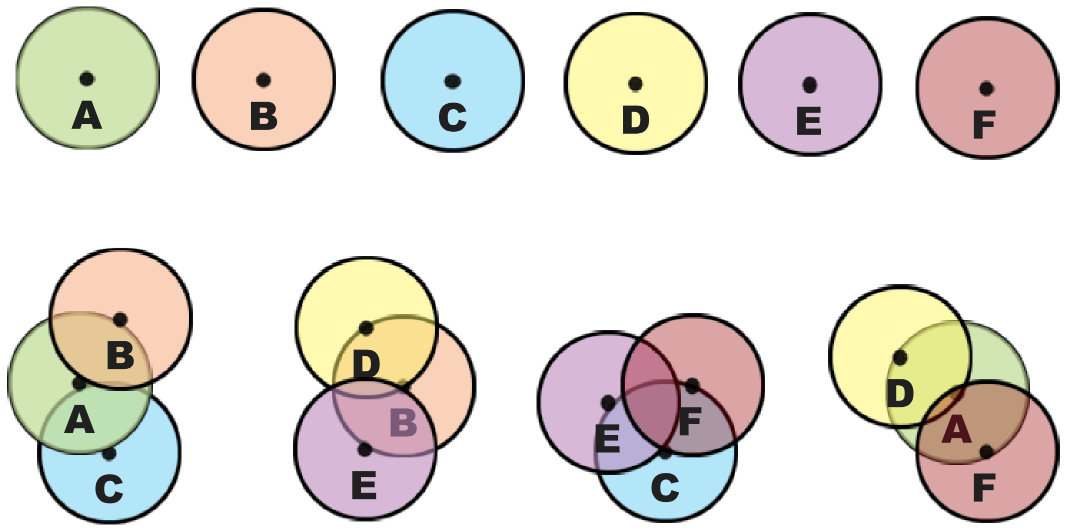
THEOREM 1. *Given a graph G, the error bound ϵ, the diameter upper bound D_{max}, the support threshold σ and K, with probability at least $1 - \epsilon$,* SpiderMine *returns a set S of top-K largest subgraphs of G such that for each $P \in S$, $|P_{sup}| \geq \sigma$ and $diam(P) \leq D_{max}$.*

Why Spiders?

(1) Reducing combinatorial complexity

Observation:

- Spiders are shared by many large patterns.
- Once obtained, they can be efficiently assembled to generate large patterns.

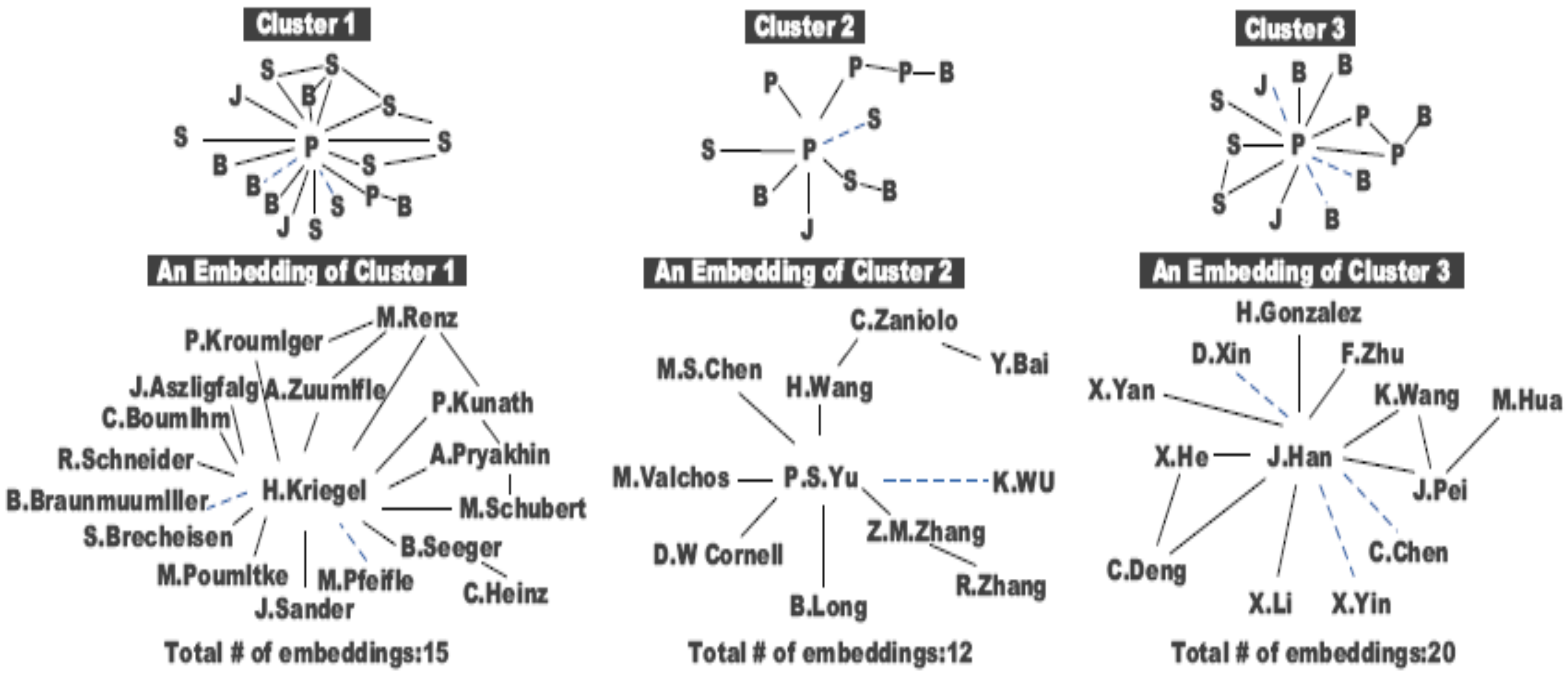


Why Spiders?

(2) Reducing graph isomorphism checking

- ❑ We propose a novel graph pattern representation --- spider-set representation.
- ❑ A pattern is represented by the set of its constituent r-spiders.
- ❑ Two same patterns must have the same spider-set representation.
- ❑ Two same spider-set representations highly likely correspond to the same pattern.
- ❑ The larger the r, the more effective the pruning.

DBLP Collaborative Patterns

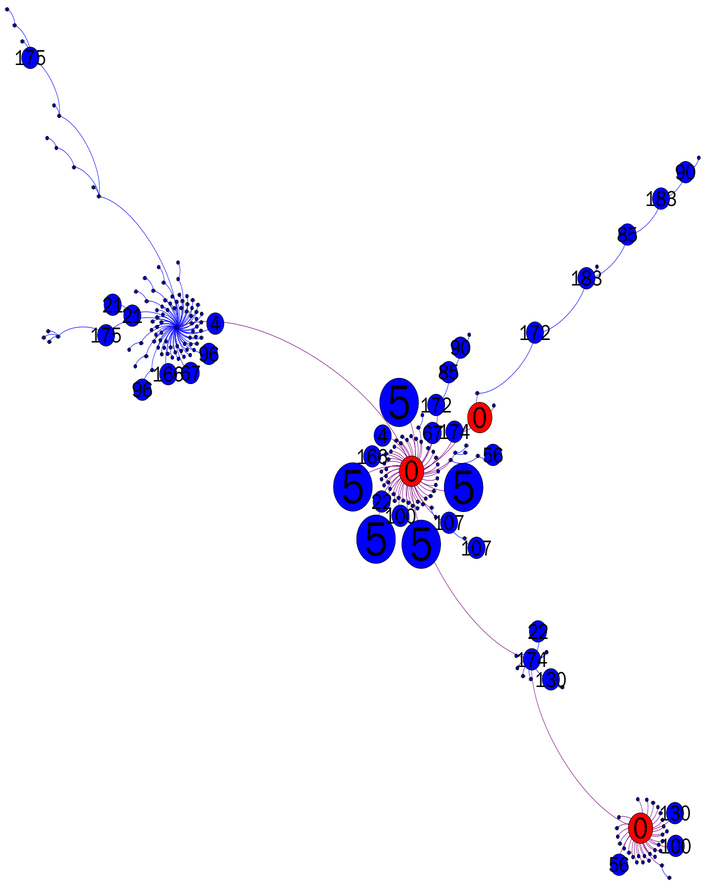


SpiderMine's response to the requirements

- Single-graph setting
  - Handles overlapping embeddings
- Huge network size
  - A probabilistic framework to capture only the top-k largest frequent patterns by leaping pattern growth.
- User-specified constraints
  - Pushed deep into the mining process by serving as the probabilistic bound

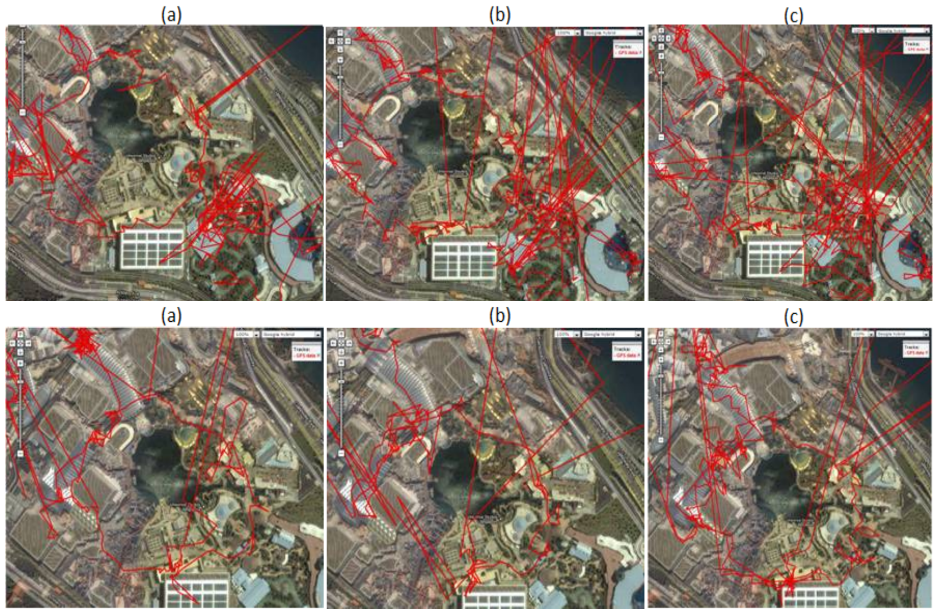
SkinnyMine --- Catching the long and skinny ones
[Zhu et al. SIGMOD'13]

- ❑Information diffusion patterns in social network
  - ❑Viral topics, rumors, etc.
  - ❑Diffusion paths are long backbones, associated ego-networks and local communities are of small radius



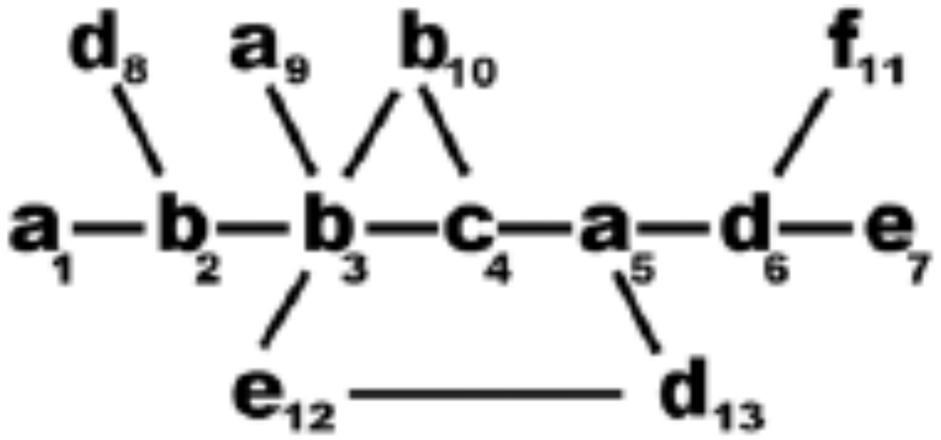
A Motivating Example

- ❑Mobile trajectory patterns in LBS and Check-in data
  - ❑Trajectories are long backbones
  - ❑Associated point of interest



Skinny Graph Patterns

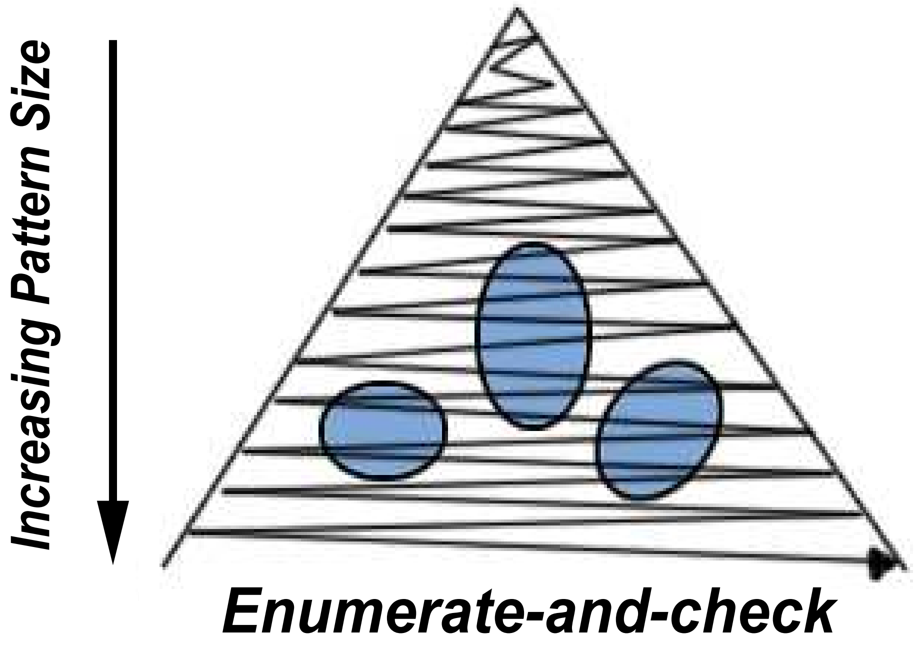
DEFINITION 8. **[l-Long δ-Skinny Pattern Mining]** *Given a graph G, a frequency threshold σ, a length l and a skinniness bound δ, the problem of* l-Long δ-Skinny Pattern Mining ((l, δ)-SPM) *is to find all l-long δ-skinny subgraphs P of G such that $|E[P]| \geq \sigma$.*



A 6-long 1-skinny pattern

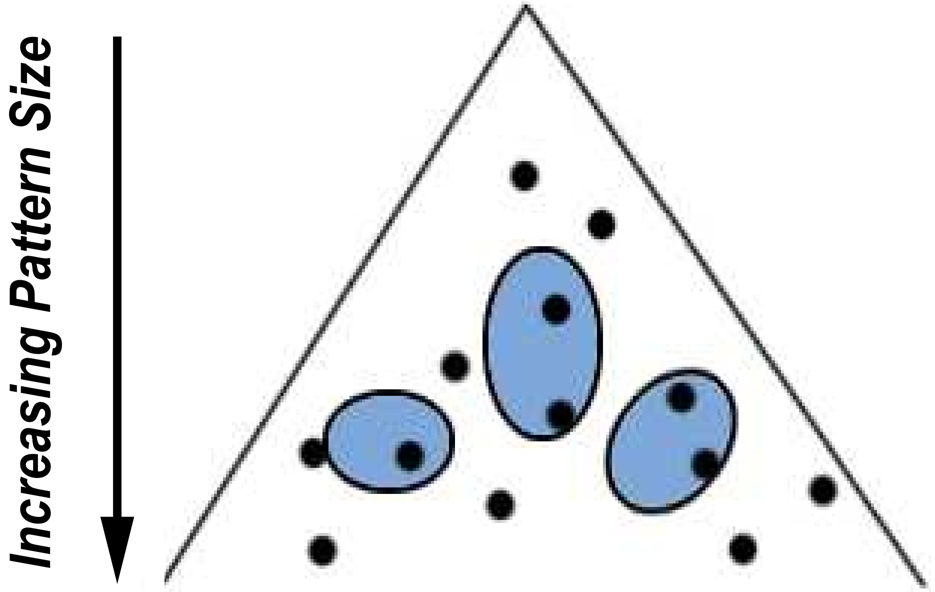
Existing Mining Paradigm I

- ❑Enumerate-and-Check
  - ❑Exhaustive enumeration of all frequent patterns and check if constraints are satisfied
  - ❑Complete pattern result
  - ❑Computationally unaffordable due to pattern explosion
  - ❑Representative algorithms
    - ❑gSpan, AGM, FSG,
    - ❑Borgelt and Berthold, FFSM, etc.



Existing Mining Paradigm II

- ❑Output Space Sampling
  - ❑A representative random sample of all frequent patterns and check if constraints are satisfied
  - ❑Efficient
  - ❑Returns only a sparse subset of the target result set
- ❑Representative algorithms
  - ❑ORIGAMI, SpiderMine, etc.

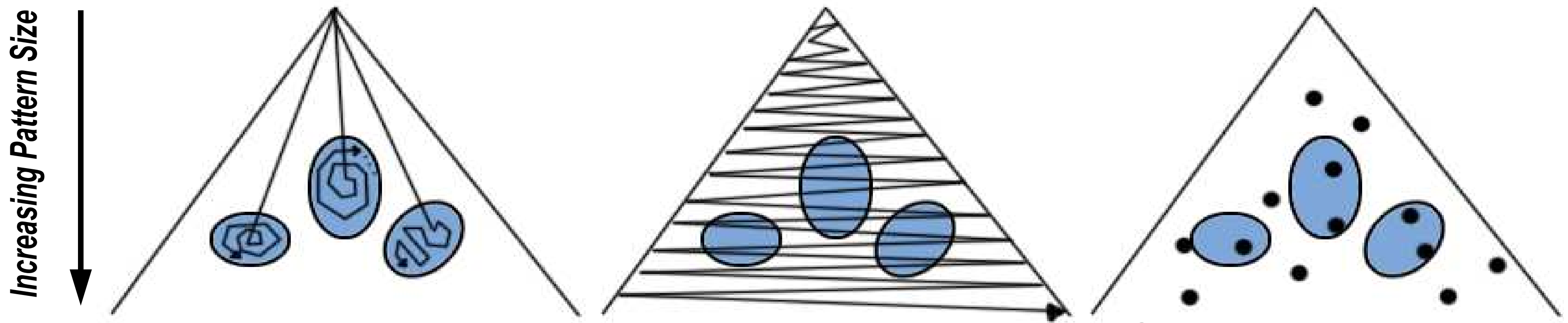


Output space sampling

Paradigm Comparison

- What we actually need
  - Complete result
  - Affordable computation time
  - Agile mining in terms of constraints
- A mining framework that is “direct”, “light” and “precise”

A pattern lattice perspective for constrained frequent pattern mining

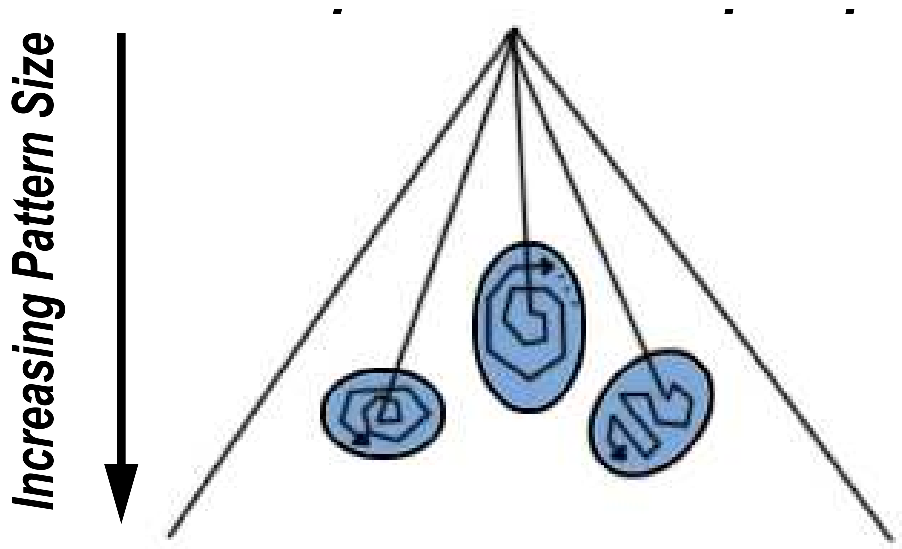


Direct mining: SkinnyMine *Enumerate-and-check* *Output space sampling*

Design Ideas

- ❑Minimal Constraint-Satisfying Pattern
  - ❑Easy to compute with a low cost
- ❑Constraint-Preserving Pattern Growth
  - ❑Non-redundant pattern generation

- ❑ How to generate each skinny pattern exactly once?
- ❑ Canonical diameters



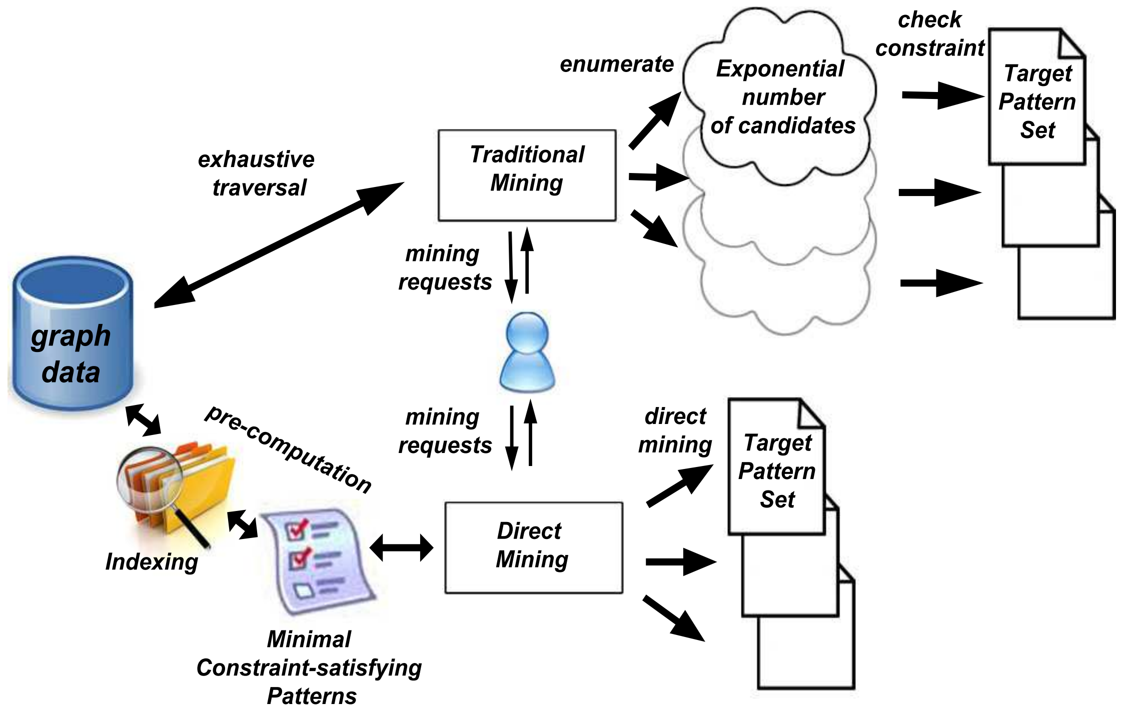
Direct mining: SkinnyMine

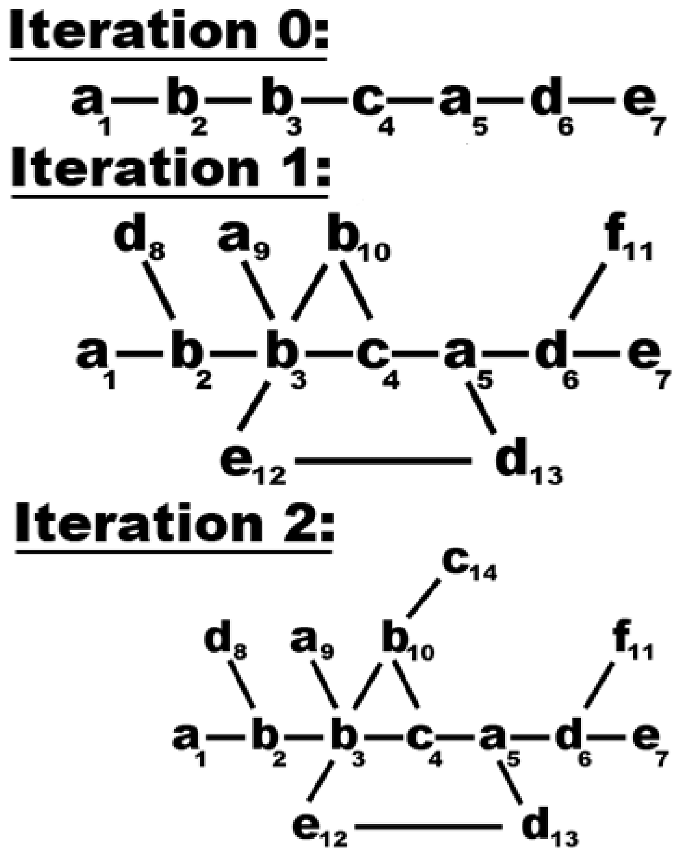
Canonical Diameters

- ❑Each pattern has a unique canonical diameter.
- ❑The complete result set is partitioned by canonical diameters.
- ❑Patterns can be generated uniquely if they are generated only from the set corresponding to their canonical diameters.

Algorithm Overview

- ❑Two stages
  - ❑Stage I: Mining canonical diameters
  - ❑Stage II: Growing canonical diameters to patterns by level.



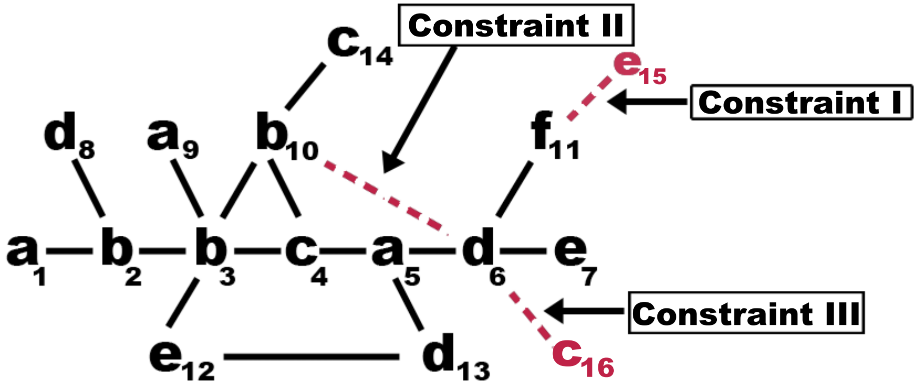


Maintaining Canonical Diameters

1. **Constraint I: Diameter is not increased.**
   This means $\mathbb{D}(P') \leq \mathbb{D}(P)$, which guarantees that the edge extension does not create a longer diameter.

2. **Constraint II: $\mathbb{L}$ still realizes the shortest distance between v_H and v_T.**
   This means in P', $Dist(v_H, v_T) = |\mathbb{L}|$ and therefore $\mathbb{D}(P') \geq \mathbb{D}(P)$, which guarantees that the edge extension does not shorten the distance between v_H and v_T.

3. **Constraint III: $\mathbb{L} \prec \mathbb{L}'$ for any newly generated diameter $\mathbb{L}'$ of the same length.**

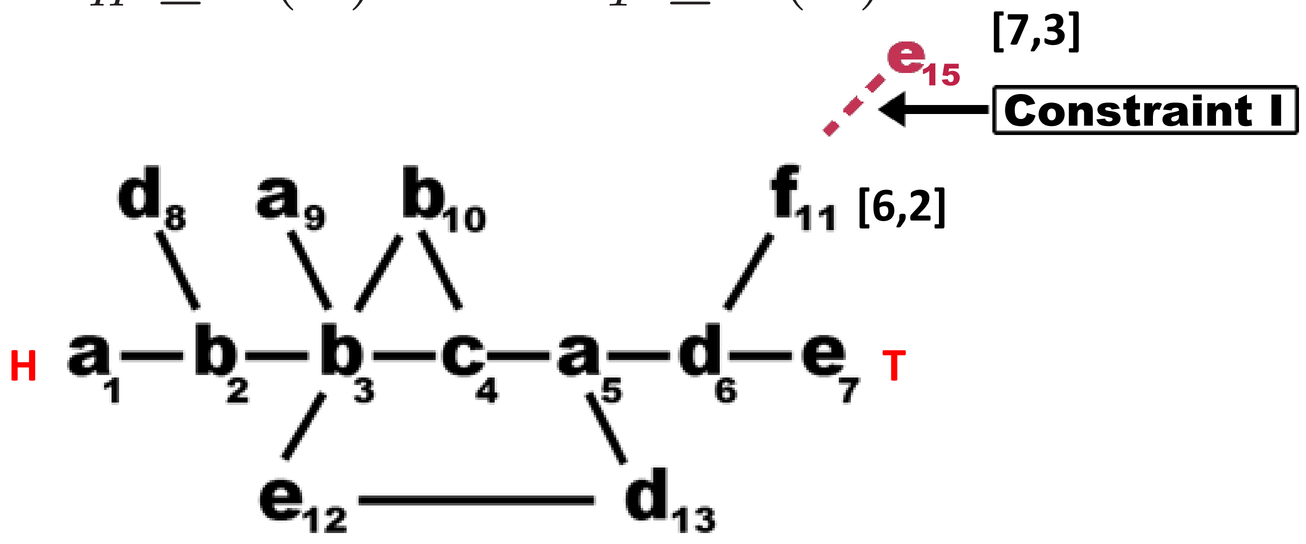


Efficient Constraint Maintenance

- For each node v on the current growth frontier, we maintain two distances D^v_H and D^v_T. When adding new edge or new node u, we only need to check these two distances.

Constraint I: Diameter is not increased

$$D_H^u \leq \mathbb{D}(P) \ \ and \ \ D_T^u \leq \mathbb{D}(P)$$



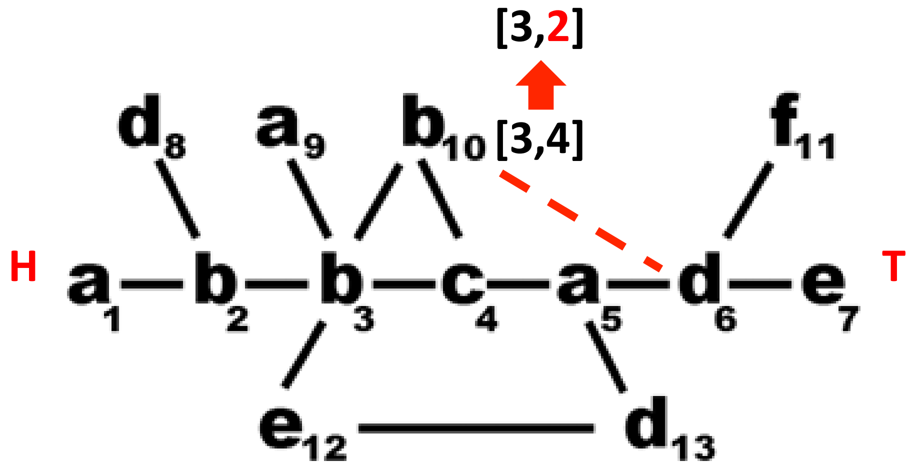
D(P)=6

Efficient Constraint Maintenance

- ❑For each node v on the current growth frontier, we maintain two distances D^v_H and D^v_T. When adding new edge or new node u, we only need to check these two distances.

Constraint II: The shortest distance between H and T has not been shortened

$$D_H^u + D_T^u \geq \mathbb{D}(P)$$



D(P)=6

A General Direct Mining Framework

❑Two essential properties for constraints

PROPERTY 1. ***[Reducibility]*** *A graph constraint C is called* reducible *if there exists a positive integer k and a nonempty set S of graph patterns such that for each $P \in S$, $|E(P)| \geq k$ and $f_C(P) = 1$, and for any pattern $P' \subset P$, $f_C(P') = 0$.*

PROPERTY 2. ***[Continuity]*** *A graph constraint C is called* continuous *if for each pattern P such that $f_C(P) = 1$, one of the following is true: (1)P is a minimal constraint-satisfying pattern; (2)there exists at least one pattern P' such that $P' \subset P$, $|E(P')| = |E(P)| - 1$ and $f_C(P') = 1$.*

DBLP Collaborative Patterns

- Real Data: DBLP

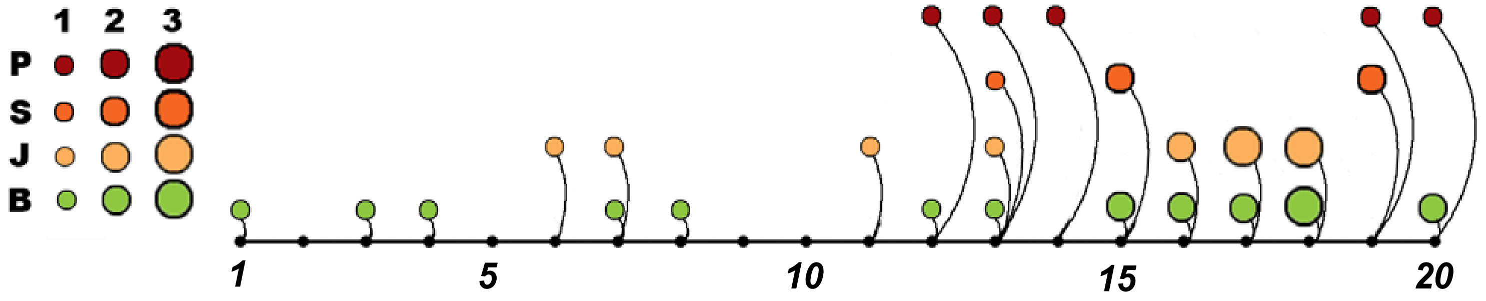
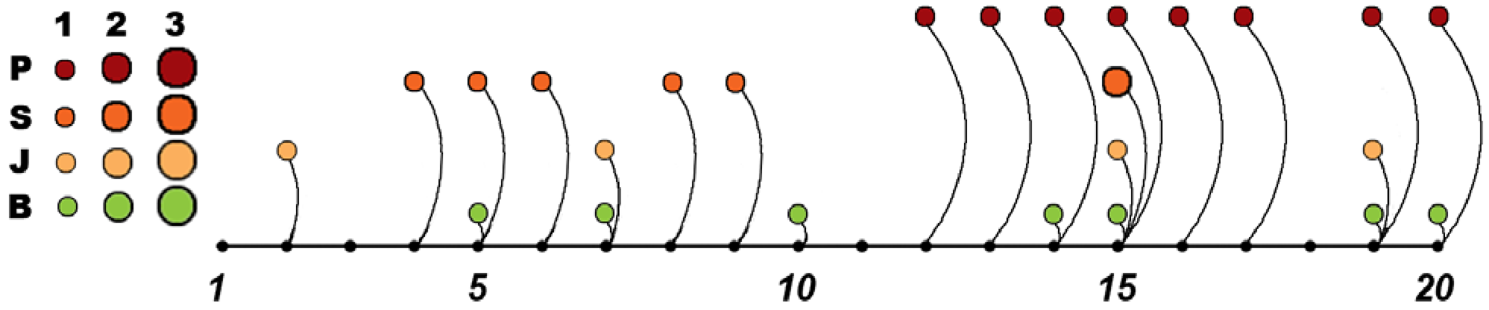
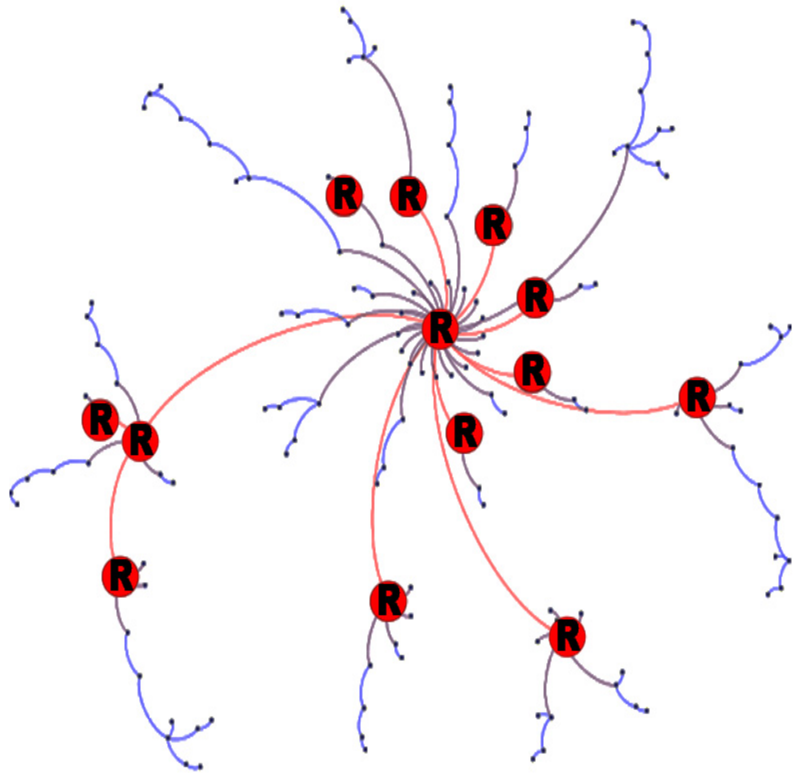


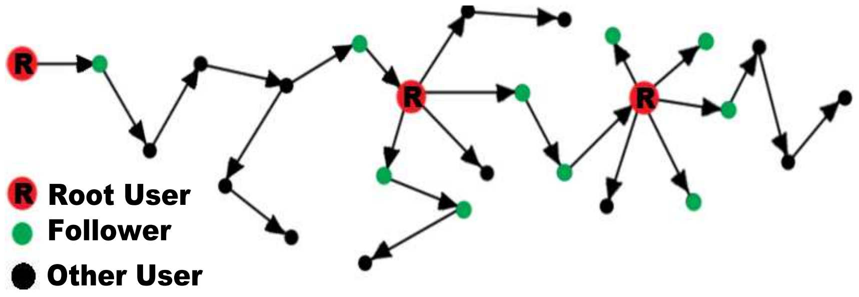
Figure 21: DBLP skinny pattern example 1: A temporal collaborative pattern across 20 years.



Information Diffusion Patterns

- Real Data: Sina Weibo





SkinnyMine's response to the requirements

- Single-graph setting
  - Handles overlapping embeddings
- Huge network size
  - A direct mining strategy to reach target patterns as quickly as possible, and locally identify only those of interest
- User-specified constraints
  - Pushed deep into the mining process by guaranteeing the non-redundant pattern generation

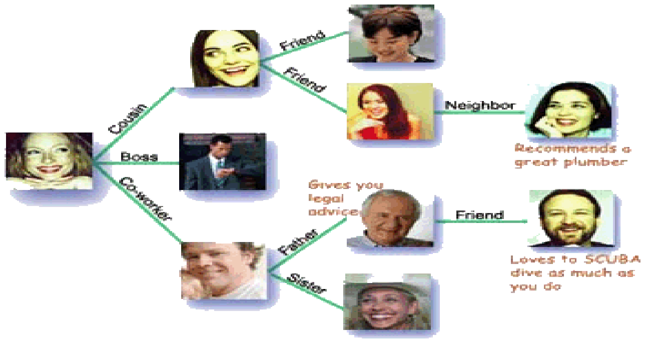
References

- F. Zhu, Q. Qu, D. Lo, X. Yan, J. Han, and P. S. Yu, *"**Mining**
- ***top-k large structural patterns in a massive network**"*, PVLDB, 2011.
- F. Zhu, Z. Zhang and Q,Qu, ***"A Direct Mining Approach To Efficient Constrained Graph Pattern Discovery"***. SIGMOD 2013.
- F. Zhu, X. Yan, J. Han, and P. Yu. ***"gPrune: A constraint pushing framework for graph pattern mining"***, In PAKDD, 2007.
- M. A. Hasan, V. Chaoji, S. Salem, J. Besson, and M. J. Zaki**.** ***"Origami: Mining representative orthogonal graph patterns"***. In ICDM 2007.

Part III
Collaboration Patterns

Collaborations Are Everywhere

Question answering in a social network



Customer service

Collaborator finding in the academic world

Business referral

Software development

Collaborations In This Talk

- ❑ **Peer Interaction**
  - o Patterns
  - o Prediction
  - o Optimization

- ❑ **Team Joint Collaboration**
  - o Patterns
  - o Prediction
  - o Optimization

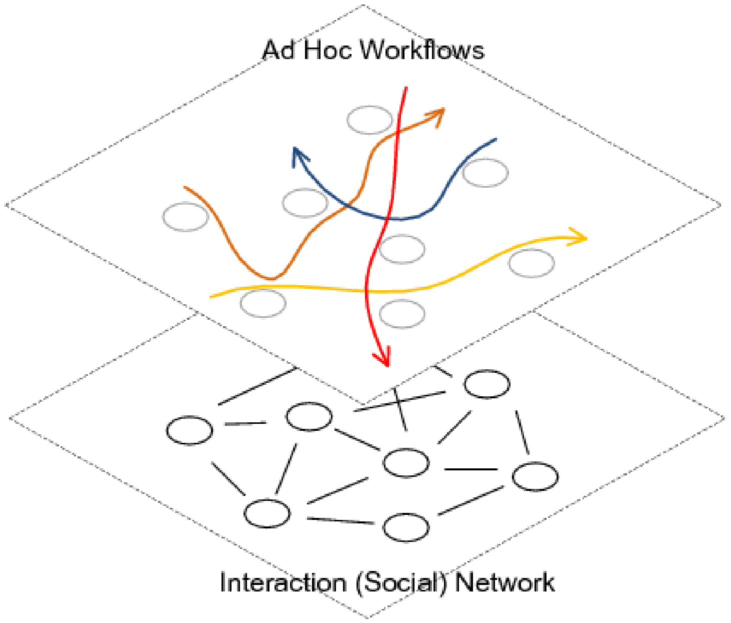
Collaborations In This Talk

❑ **Peer Interaction**
- o Patterns
- o Prediction
- o Optimization

❑ **Team Joint Collaboration**
- o Patterns
- o Prediction
- o Optimization

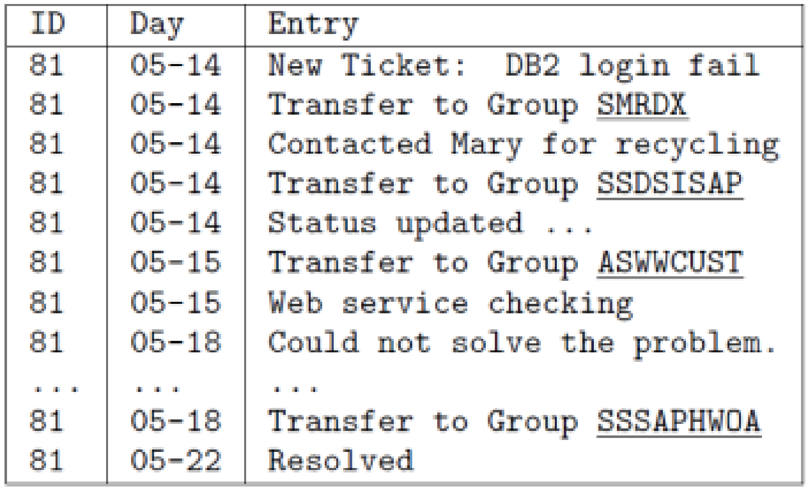
Collaborative Networks

Information Flow



Collaborative Network

Enterprise problem ticket



ID	Day	Entry
81	05-14	New Ticket: DB2 login fail
81	05-14	Transfer to Group SMRDX
81	05-14	Contacted Mary for recycling
81	05-14	Transfer to Group SSDSISAP
81	05-14	Status updated ...
81	05-15	Transfer to Group ASWWCUST
81	05-15	Web service checking
81	05-18	Could not solve the problem.
...	...	...
81	05-18	Transfer to Group SSSAPHWOA
81	05-22	Resolved

Eclipse bug record

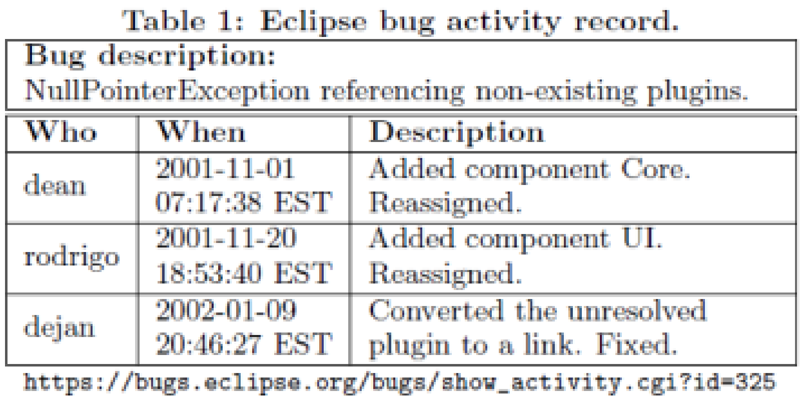


Table 1: Eclipse bug activity record.

Bug description:
NullPointerException referencing non-existing plugins.

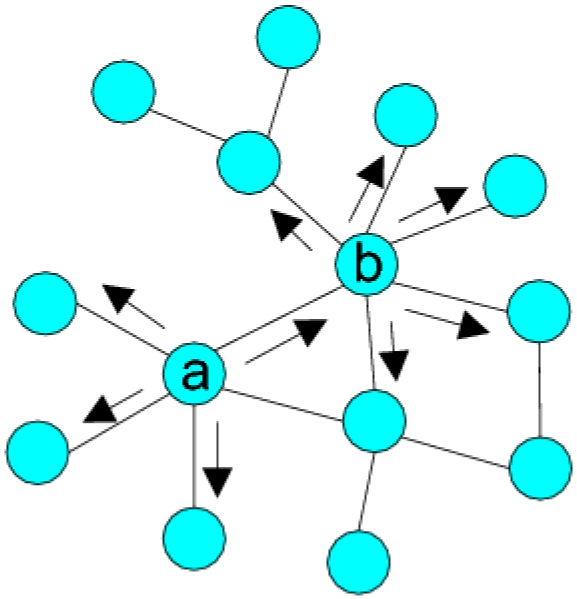
Who	When	Description
dean	2001-11-01 07:17:38 EST	Added component Core. Reassigned.
rodrigo	2001-11-20 18:53:40 EST	Added component UI. Reassigned.
dejan	2002-01-09 20:46:27 EST	Converted the unresolved plugin to a link. Fixed.

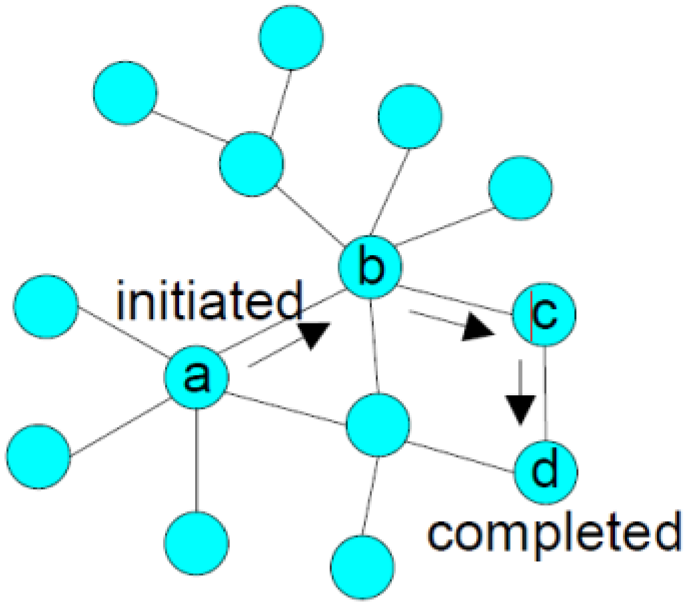
https://bugs.eclipse.org/bugs/show_activity.cgi?id=325

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

Social Networks VS. Collaborative Networks





Traditional Social Networks

(Facebook, Twitter …)

• Non-task driven information diffusion

Collaborative Networks

(problem solving, team work ...)

•Task-driven information flow

We want to analyze…

- **How do experts make routing decisions?**
- **Who have made inefficient routing decisions?**
- **How to optimize the routing performance through targeted training?**
- **Can the completion time of a task be predicted so that one can act early for difficult tasks?**

In order to understand human factors in real task routing and the inefficiency bottleneck, take early actions, and further improve real collaborative networks.

We want to analyze…

- How do experts make routing decisions?
- Who have made inefficient routing decisions?
- How to optimize the routing performance through targeted training?
- Can the completion time of a task be predicted so that one can act early for difficult tasks?

If we use intelligent algorithms to replace humans,

- **Can we predict next best experts?**
- **How to model the collaborative networks?**

We want to analyze…

- How do experts make routing decisions?
- Who have made inefficient routing decisions?
- How to optimize the routing performance through targeted training?
- Can the completion time of a task be predicted so that one can act early for difficult tasks?
- Can we predict next best experts?
- How to model the collaborative networks?

[Sun et al., KDD'14]

Observations on Routing Decision Making

Observation 1

Tasks with similar content, but different routing sequences

e.g., two problem tickets in IBM IT service department

Task ID	Task Content	Routing Sequence
492	Need password reset for kasperj on machine "pathfinder"". Route to NUS_N_DCRCHAIX	$12 \rightarrow 505 \rightarrow 1914 \rightarrow 1915 \rightarrow 1916 \rightarrow 247$
494	Need password reset for jhallacy on machine ""pathfinder"". Route to NUS_N_DCRCHAIX	$12 \rightarrow 13 \rightarrow 86$

Routing decision is not deterministic, given a certain task.

UCSB

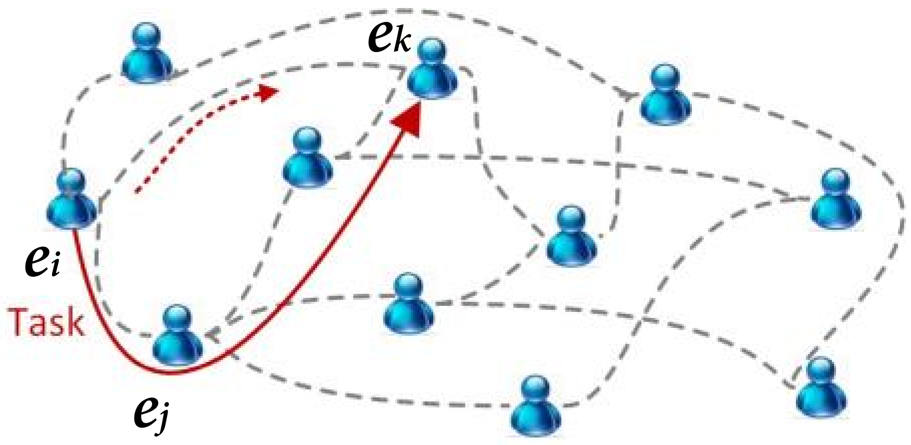
SMU SINGAPORE MANAGEMENT UNIVERSITY

[Sun et al., KDD'14]

Observations on Routing Decision Making

Observation 2

An expert might not directly send a task to a resolver.



Is it because he does not understand the task very well, thus randomly routing it? Or

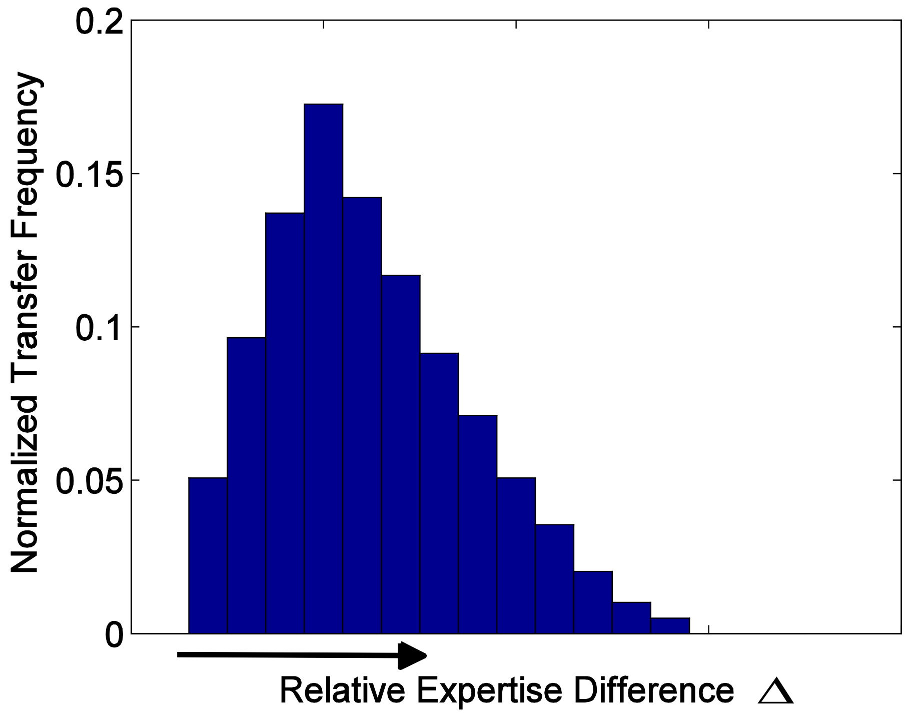
he believes the other expert has a better chance to solve it, or a better chance to find the right expert to solve it?

[Sun et al., KDD'14]

Observations on Routing Decision Making

Observation 3

An expert tends to transfer a task to some expert whose expertise is *neither* too close *nor* too far from his own, i.e., not necessarily the final resolver!



log-normal density:

$$f \propto \frac{1}{\Delta} e^{-\frac{[\ln \Delta - \mu]^2}{2\sigma^2}}$$

[Sun et al., KDD'14]

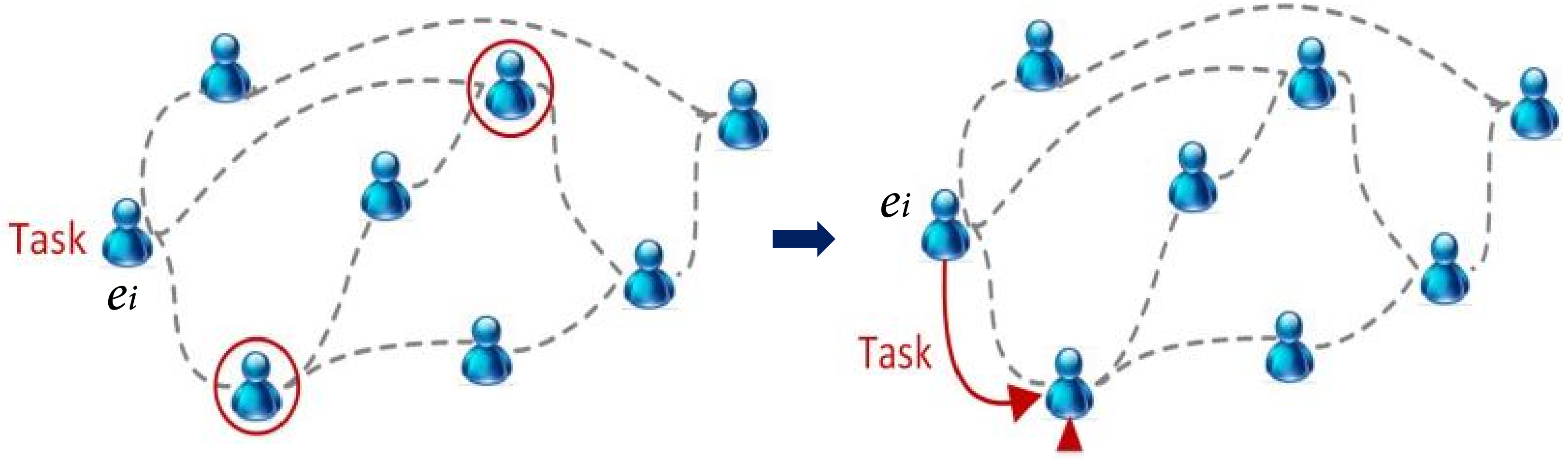
Modeling Expert Decision Logic

- A Two-Phase Assumption

When an expert transfers a task,

Phase I

Phase II



Candidate pool establishment

Specific expert selection

[Sun et al., KDD'14]

Modeling Expert Decision Logic

- 6 particular routing patterns

Phase I

Phase II

	Task-Neutral Routing (TNR)	Task-Specific Routing (TSR)
Uniform Random (UR)	TNR^{ur}	TSR^{ur}
Volume-biased Random (VR)	TNR^{vr}	TSR^{vr}
Expertise Difference (EX)	TNR^{ex}	TSR^{ex}

[Sun et al., KDD'14]

Modeling Expert Decision Logic

- Generative model

Assume: the routing decision of an expert is generated through a mixture of 6 routing patterns.

Decision generation process:

For each expert e_i to transfer tasks,

-Draw the mixture weights of 6 routing patterns:

$$\theta_i \sim \text{Dirichlet}(\alpha)$$

(reflecting e_i's preferences over adopting different routing patterns)

-For each task t to be transferred by expert e_i,

* Draw a pattern label: $Z_{i,t} \sim \text{Mult}(\theta_i)$

* Draw an expert from the candidate pool to receive t.

[Sun et al., KDD'14]

Evaluation

- Evaluation Measure
  - Completion Time (CT): Number of experts contacted before a task is resolved
  - Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{|\text{Test Set}|} \sum_{t \in \text{Test Set}} |\widehat{\mathrm{CT}}_t - \mathrm{CT}_t|$$

- Results

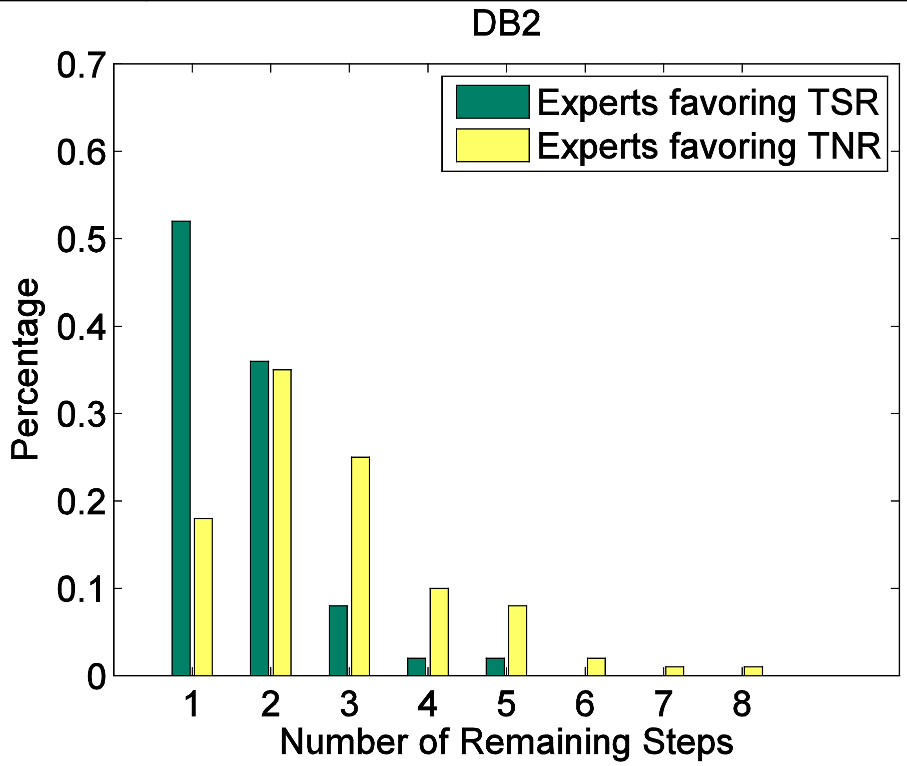
DB2	
Models	**MAE**
TNR+TSR	**0.08**
Miao *et al*.	0.68
Support Vector Regression	0.80
Bayesian regression	0.84

[Sun et al., KDD'14]

Experiments

- Resolution Efficiency of Routing Patterns TNR vs. TSR

Experts favoring TNR	Weight on TNR patterns $>$ Weight on TSR patterns.
Experts favoring TSR	Weight on TNR patterns $<$ Weight on TSR patterns



[Sun et al., KDD'14]

Application: Optimizing Collaborations

- Which expert should be trained first to adopt the most efficient routing patterns? How much efficiency improvement can we expect?
  - Random: random selection
  - Frequent transferor: select the expert who transfers the most tasks
  - Least efficient: select the least efficient expert

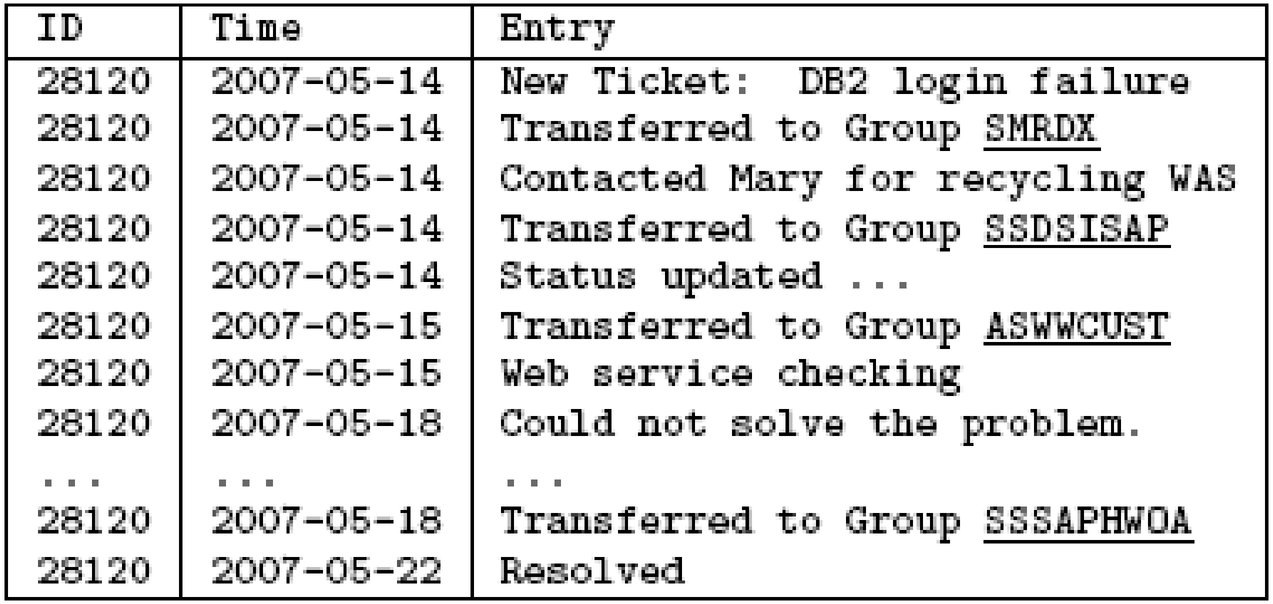
Methods	Efficiency Improvement (%)
Random	0.27
Frequent Transferor	0.91
Least Efficient	1.21
Recommendation using our model	2.75

We want to analyze…

- How do experts make routing decisions? ✓
- Who have made inefficient routing decisions? ✓
- How to optimize the routing performance through targeted training? ✓
- Can the completion time of a task be predicted so that one can act early for difficult tasks? ✓
- Can we predict next best experts?
- How to model the collaborative networks?

[Shao et al., KDD'09]

Ticket Resolution



ID	Time	Entry
28120	2007-05-14	New Ticket: DB2 login failure
28120	2007-05-14	Transferred to Group SMRDX
28120	2007-05-14	Contacted Mary for recycling WAS
28120	2007-05-14	Transferred to Group SSDSISAP
28120	2007-05-14	Status updated ...
28120	2007-05-15	Transferred to Group ASWWCUST
28120	2007-05-15	Web service checking
28120	2007-05-18	Could not solve the problem.
...	...	...
28120	2007-05-18	Transferred to Group SSSAPHWOA
28120	2007-05-22	Resolved

- ❑ Managing problem tickets is a key issue in IT service industry.
- ❑ The efficiency of solving tickets highly depends on how they are routed.
- ❑ How to develop efficient automated algorithms for ticket routing?

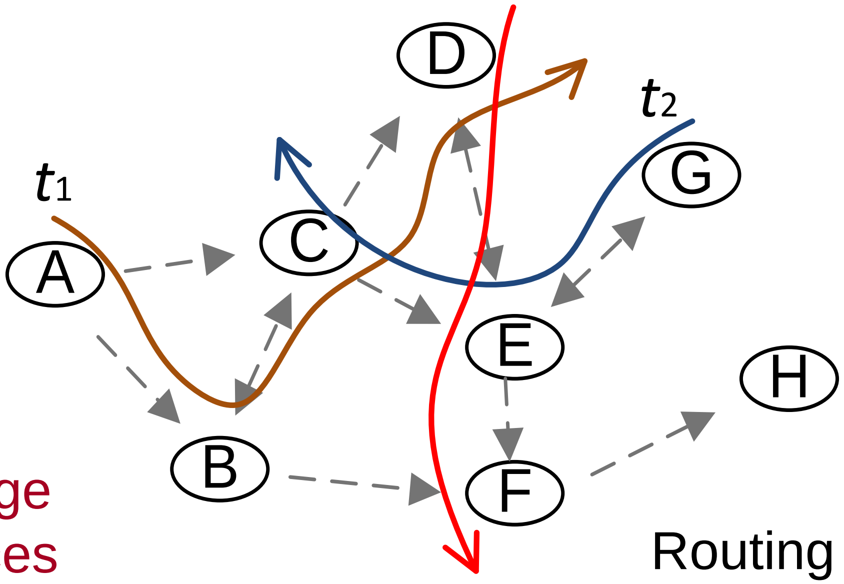
[Shao et al., KDD'08]

Automate Ticket Resolution:
Sequence Mining Approach

A set of tickets reported to the expert network $\mathcal{T} = \{t_1, t_2, ..., t_m\}$

An interconnected network of experts

$$\mathcal{G} = \{g_1, g_2, ..., g_L\}$$



Goal: Minimize the average length of routing sequences

$$S = \frac{\sum_{i=1}^{m} |R(t_i)|}{m}$$

Routing sequence of tickets

$$R(t) = g_{init}(t) -> ... -> g_{res}(t)$$

[Shao et al., KDD'08]

Markov Modeling

- ❑Capture ticket transfer patterns embedded in historical resolution sequences
  - o State: an expert group holding the ticket
  - o Transition probability: given the previous groups, the probability of the ticket being transferred to group *gi*, in the next step

$$P(g_i \mid S_{(k)}) = \begin{cases} N(g_i, S_{(k)}) / N(S_{(k)}) & \text{if } N(S_{(k)}) \\ 0 & else \end{cases}$$

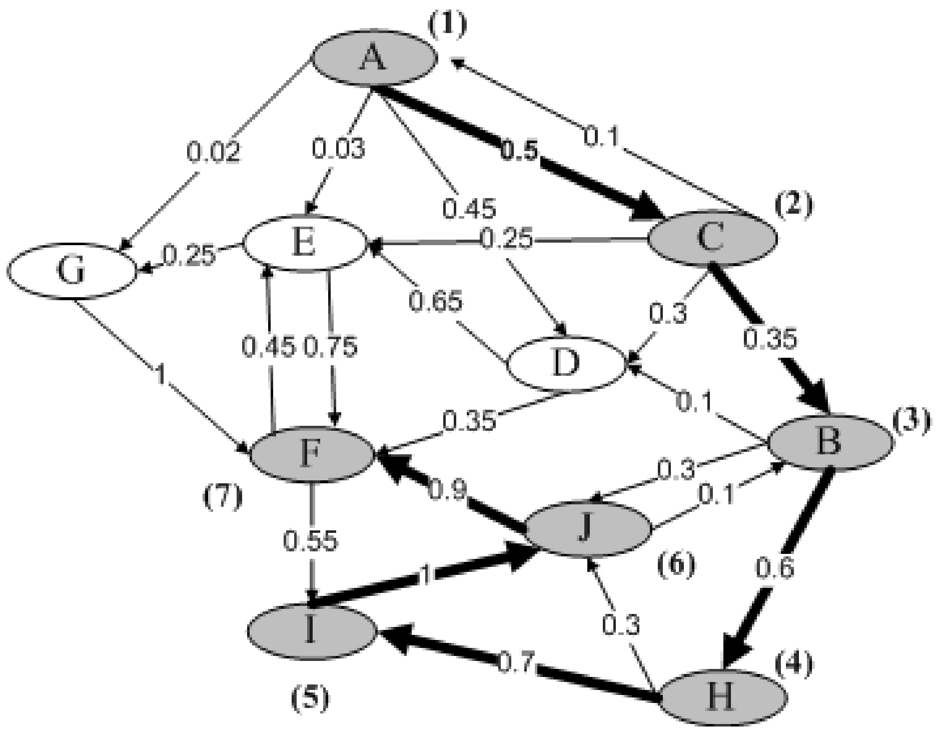
# of instances that a ticket is transferred to group *gi , after being processed by S(k)*

# of instances with a set of group transfers

[Shao et al., KDD'08]

Routing Algorithms

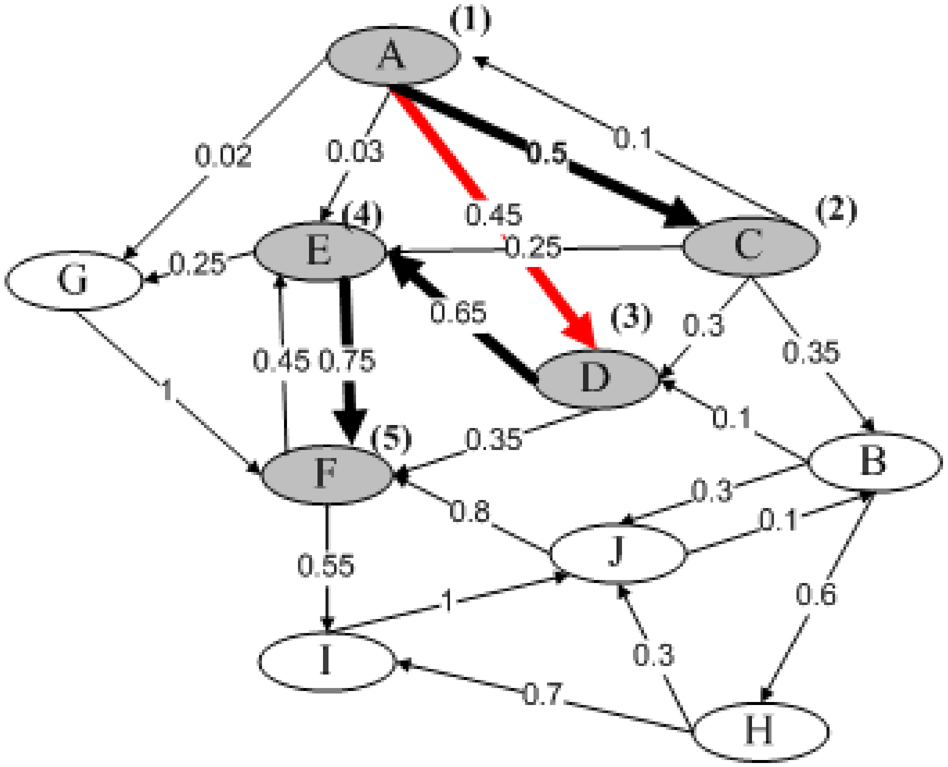
- First-order Memoryless (FM)
  - o Relies on the current state to make transfer decisions



[Shao et al., KDD'08]

Routing Algorithms

- First-order Multiple active State (FMS)
  - o Ticket transfer decision is based on any one of the past states (instead of the current state).

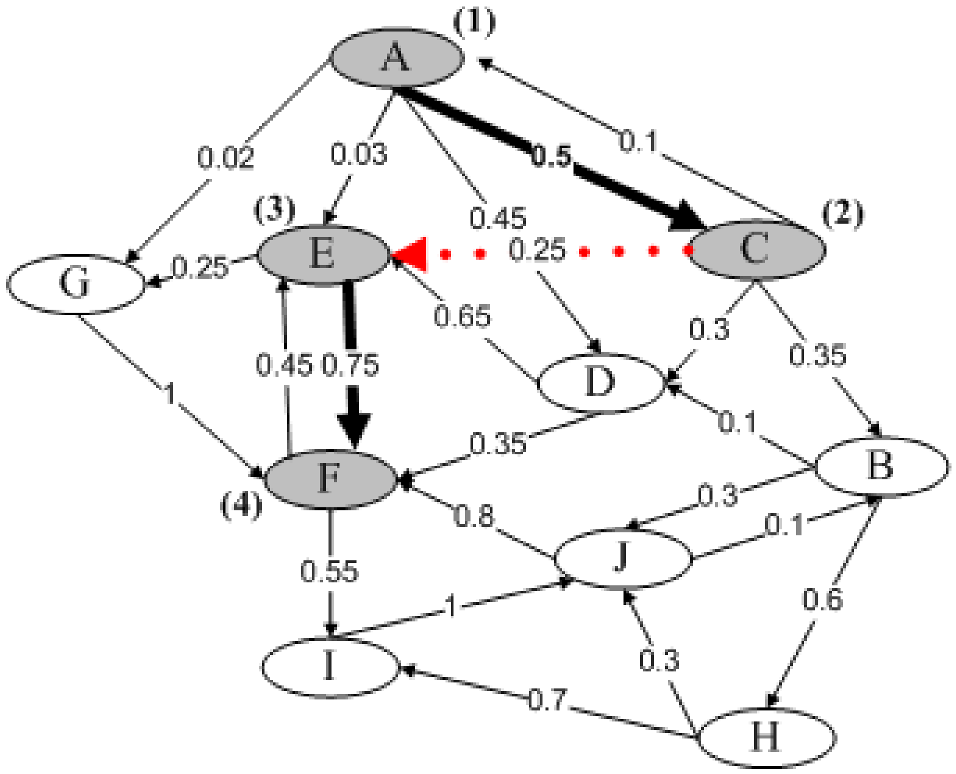


[Shao et al., KDD'08]

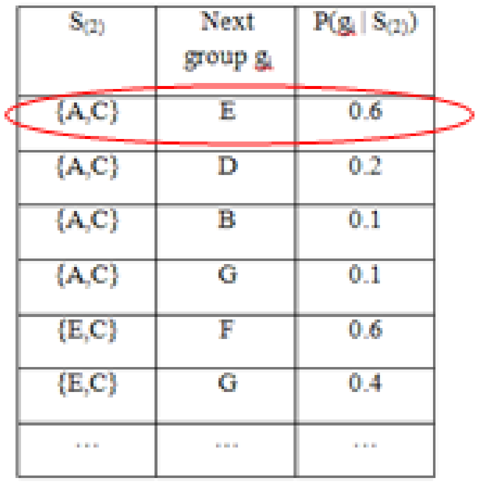
Routing Algorithms

❑Variable order Multiple active State (VMS)

- o With multiple active states, use <u>higher-order</u> whenever it can make more confident prediction



(a) Routing Steps in the model



$S_{(2)}$	Next group g_i	$P(g_i \mid S_{(2)})$
{A,C}	E	0.6
{A,C}	D	0.2
{A,C}	B	0.1
{A,C}	G	0.1
{E,C}	F	0.6
{E,C}	G	0.4
…	…	…

(b) 2nd order probability table

[Miao et al., KDD'10]

Incorporate Ticket Content:

Generative Models

A set of tickets reported to the expert network

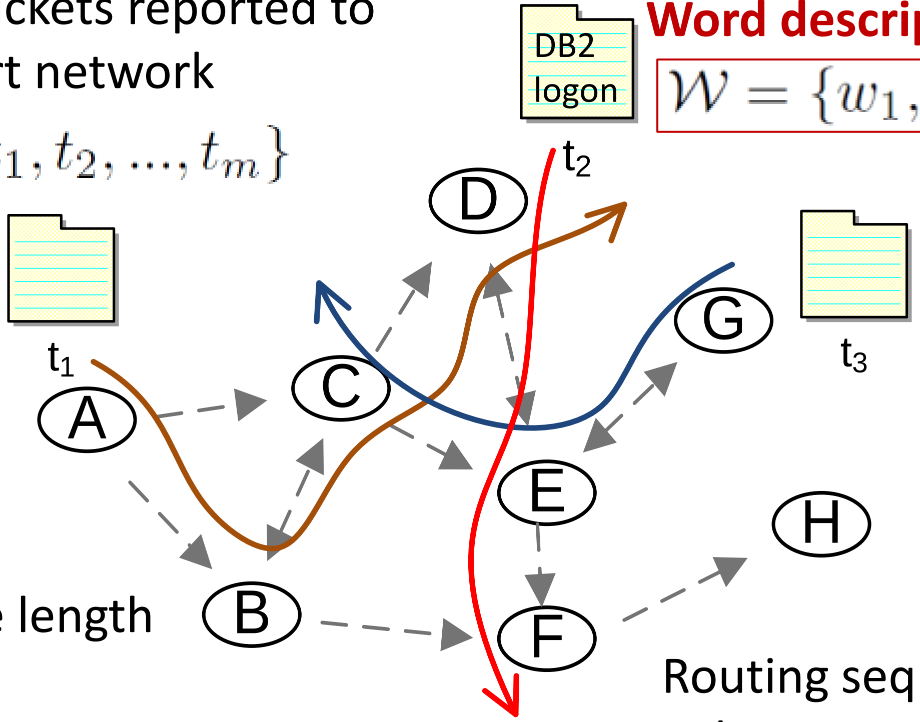
$$\mathcal{T} = \{t_1, t_2, ..., t_m\}$$

Word description of tickets

$$\boxed{\mathcal{W} = \{w_1, w_2, ..., w_n\}}$$

An interconnected network of experts

$$\mathcal{G} = \{g_1, g_2, ..., g_L\}$$



Goal: Minimize the average length of routing sequences

$$S = \frac{\sum_{i=1}^{m} |R(t_i)|}{m}$$

Routing sequence of tickets

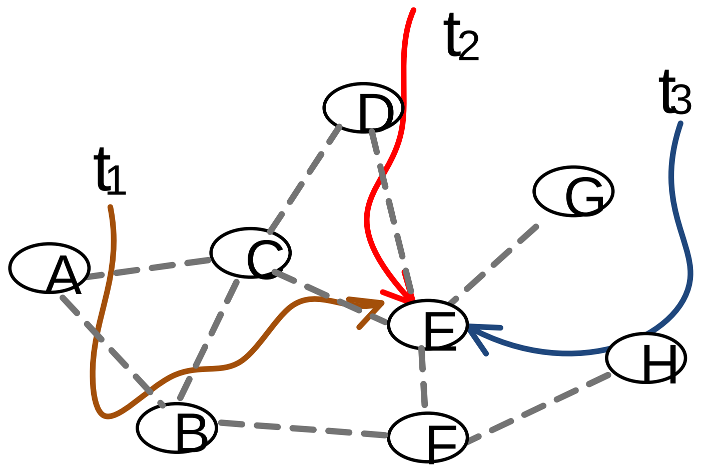
$$R(t) = g_{init}(t) -> ... -> g_{res}(t)$$

[Miao et al., KDD'10]

Resolution Model (RM)

❑Each expert has an expertise profile

- o An expert is likely to be able to resolve tickets similar to what he/she has resolved previously



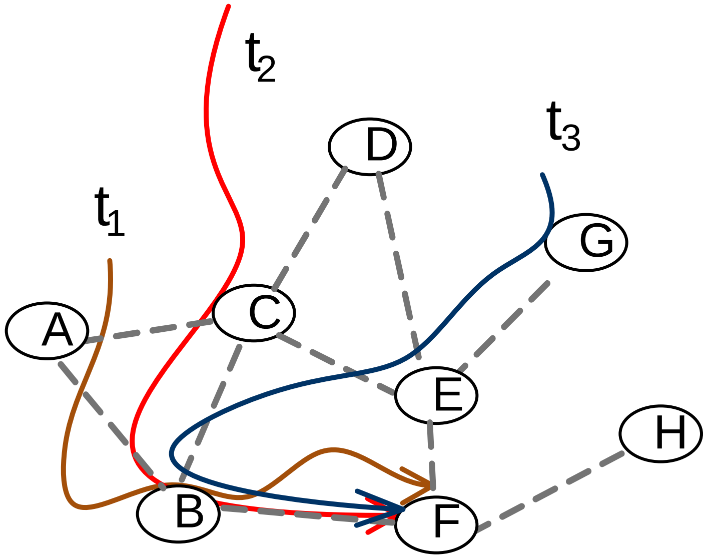
Tickets resolved by expert E

$$P_{g_i} = [P(w_1|g_i), P(w_2|g_i), ..., P(w_n|g_i)]^T$$

[Miao et al., KDD'10]

Transfer Model (TM)

- ❑Expertise awareness between experts
  - o An expert transfers similar tickets to another expert



Tickets transferred from expert B to expert F

$$P_{e_{ij}} = [P(w_1|e_{ij}), P(w_2|e_{ij}), ..., P(w_n|e_{ij})]^T$$

[Miao et al., KDD'10]

Optimized Network Model (ONM)

- Transfer profiles optimized for the entire expert network

$$\mathcal{L} = \prod_{t \in \mathcal{T}} P(R(t)|t)$$

$$P(R(t)|t) = P(g_1|t)P(g_2|t, g_1)P(g_3|t, g_2)P(g_3|t, g_3)$$

$$P(g_j|t, g_i) = \frac{P(t|e_{ij})P(g_j|g_i)}{Z(t, g_i)}$$

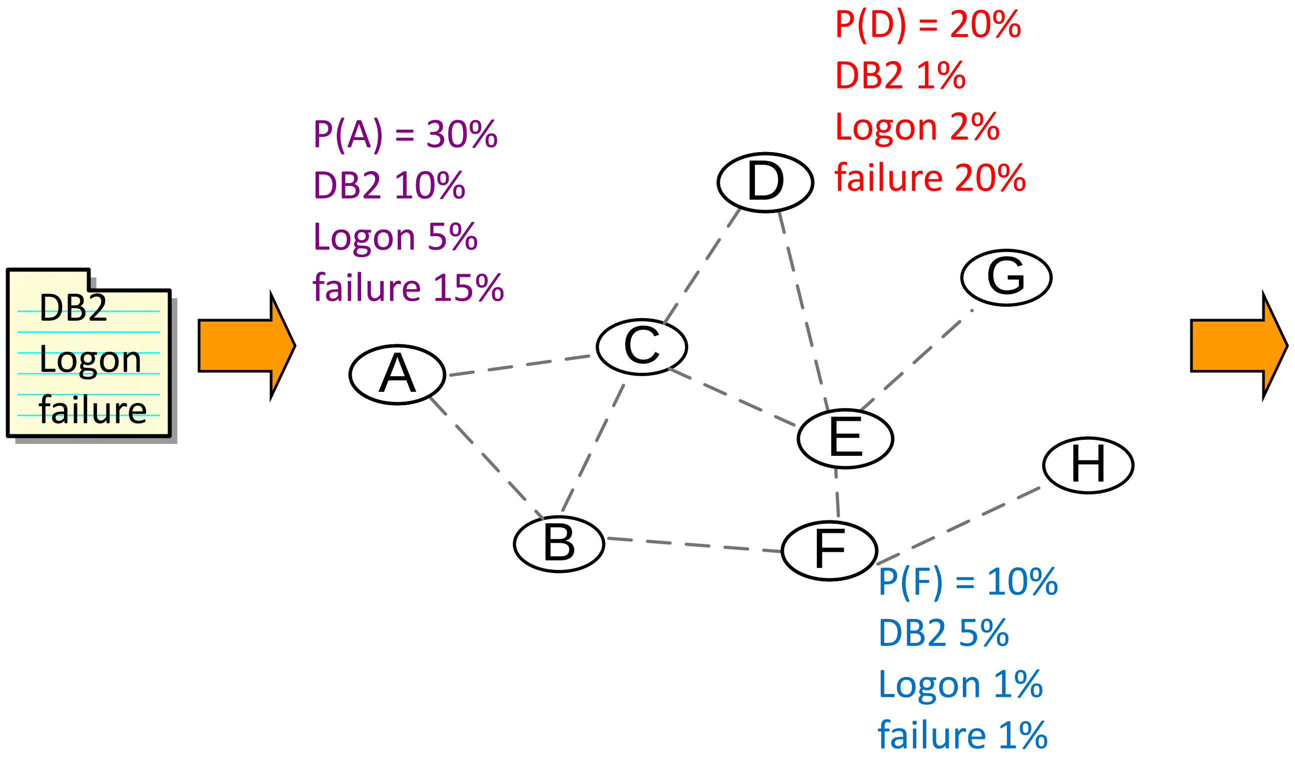
$$= \frac{(\prod_{w_k \in t} P(w_k|e_{ij})^{f(w_k, t)})P(g_j|g_i)}{Z(t, g_i)}$$

$$Z(t, g_i) = \sum_{g_j \in \mathcal{G}} P(t|e_{ij})P(g_j|g_i)$$

[Miao et al., KDD'10]

Routing Algorithms: Ranked Resolver

- Match the ticket content with the expertise profiles



A: 2.2 e-4

D: 8.0 e-6

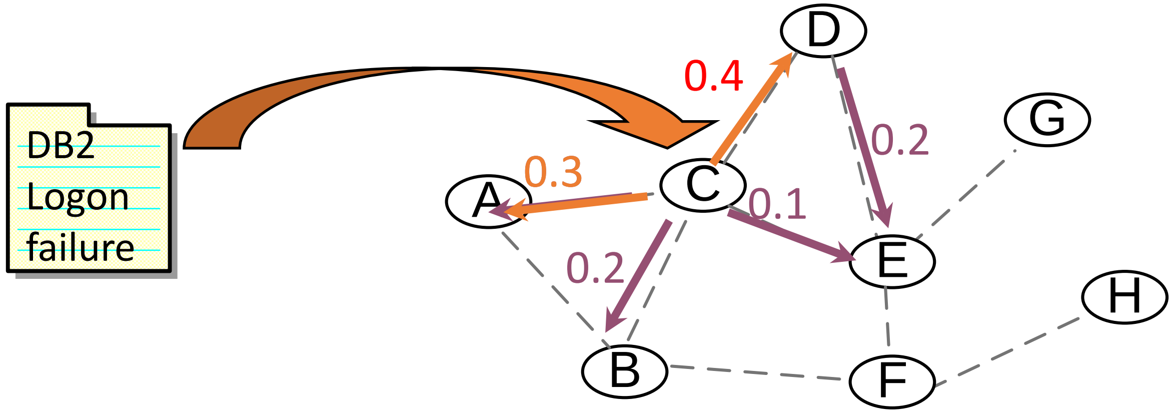
F: 5.0 e-7

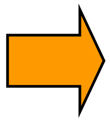
$$P(g_i|t) = \frac{P(g_i)P(t|g_i)}{P(t)} \propto P(g_i)\prod_{w_k \in t} P(w_k|g_i)^{f(w_k,t)}$$

[Miao et al., KDD'10]

Routing Algorithms: Greedy Transfer

- Match the ticket with the transfer profiles





D

A

B

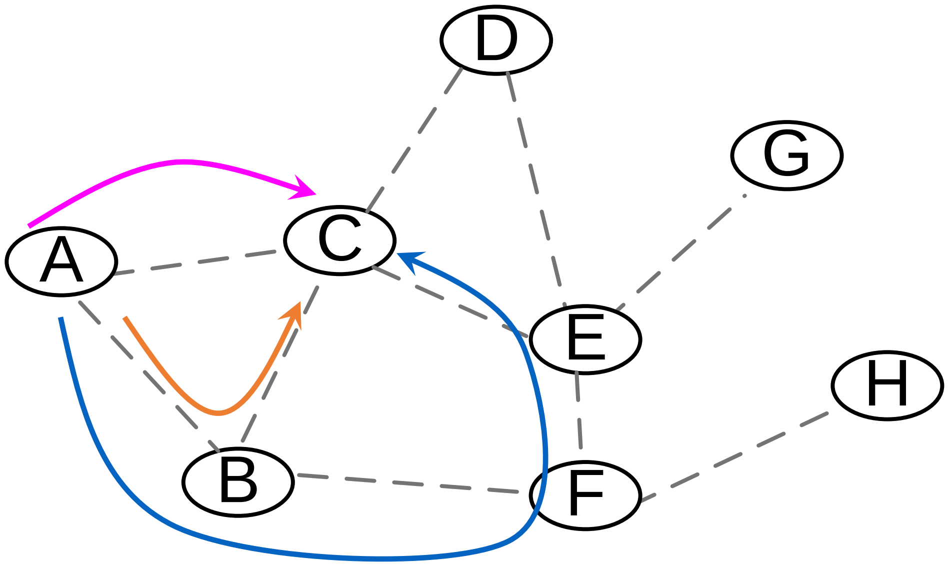
$$Rank(g_j) \propto \max_{g_i \in R(t)} P(g_j | t, g_i)$$

$$P(g_j | t, g_i) = \frac{P(g_j | g_i) \prod_{w_k \in t} P(w_k | e_{ij})^{f(w_k, t)}}{\sum_{g_l \in \mathcal{G}} P(g_l | g_i) \prod_{w_k \in t} P(w_k | e_{il})^{f(w_k, t)}}$$

[Miao et al., KDD’10]

Routing Algorithms: Holistic Routing

- All possibilities are explored

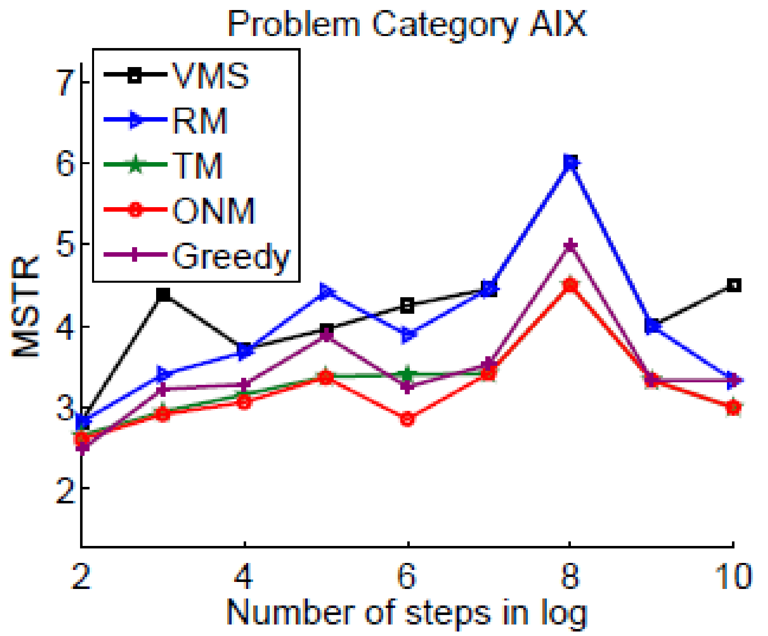


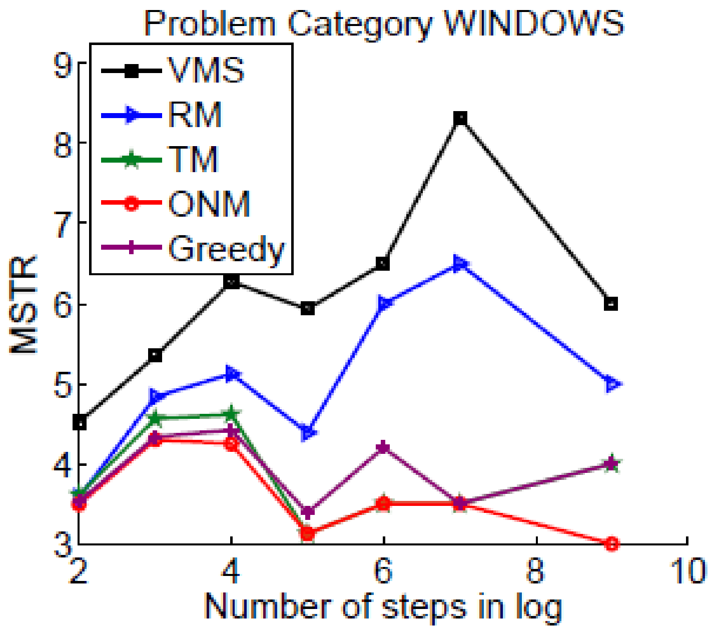
UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

[Miao et al., KDD'10]

Evaluation





We want to answer…

- How do experts make routing decisions? ✓
- Who have made inefficient routing decisions? ✓
- How to optimize the routing performance through targeted training? ✓
- Can the completion time of a task be predicted so that one can act early for difficult tasks? ✓
- Can we predict next best experts? ✓
- How to model the collaborative networks? ⬅

Collaborative Network Observations

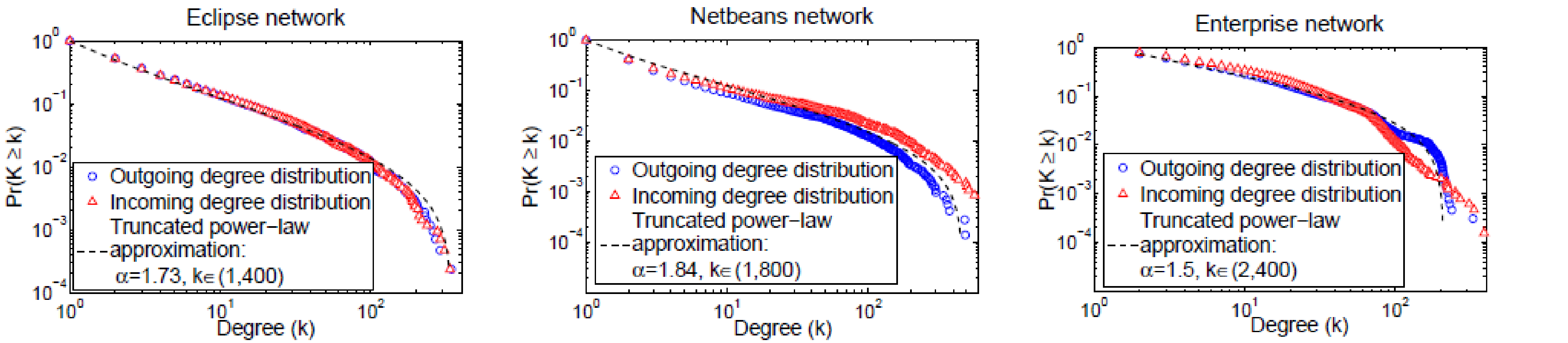


Figure 2: Degree distribution of collaborative networks.

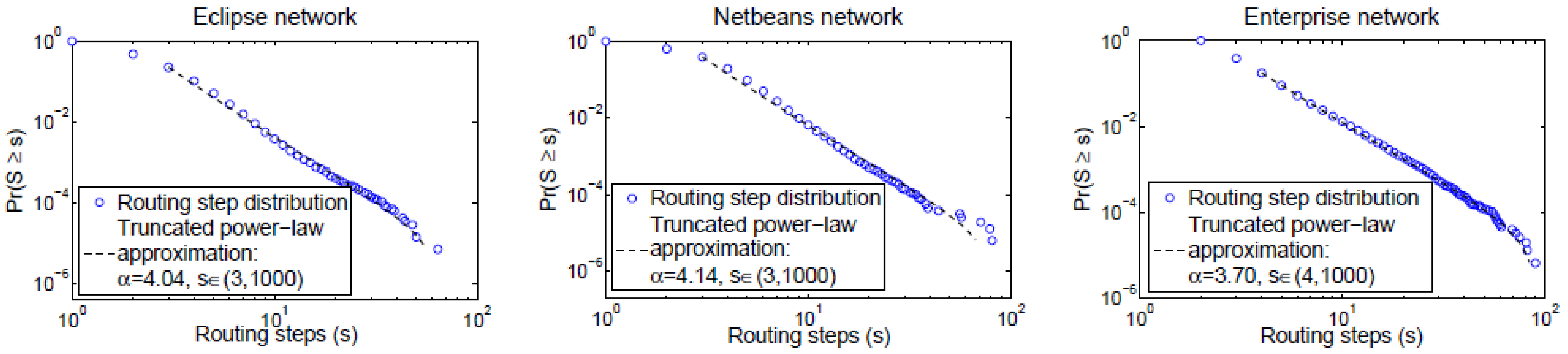


Figure 3: Routing steps distribution of problem solving in collaborative networks.

[Miao et al., WWW'12]

Collaborative Network Observations

- ❑Clustering effect
  - o Eclipse or Netbeans network
    - Developers are loosely organized
    - <u>Lower clustering coefficients</u>
  - o Enterprise network
    - Agents are more organized in groups by specialty
    - Agents inside a group is more likely to collaborate with each other
    - <u>Higher clustering coefficient</u>
- ❑Clustering coefficient of studied networks

Eclipse Network	NetBeans Network	Enterprise Network
0.198	0.21	0.35

[Miao et al., WWW’12]

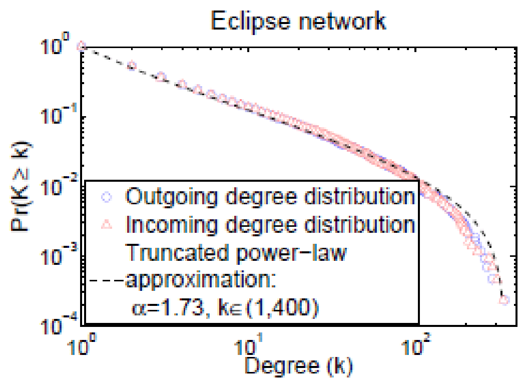
Modeling Approach

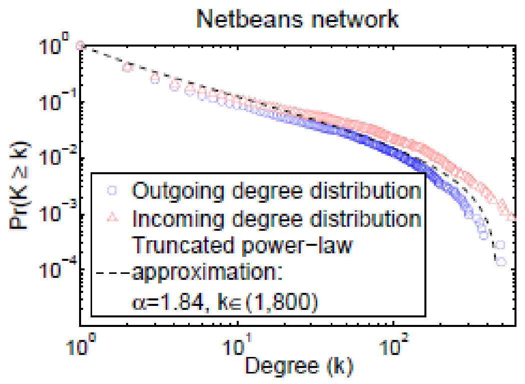
- ❑ Network Model
  - o Simulates static network connectivity
- ❑ Routing Model
  - o Simulates dynamic human behavior in information routing

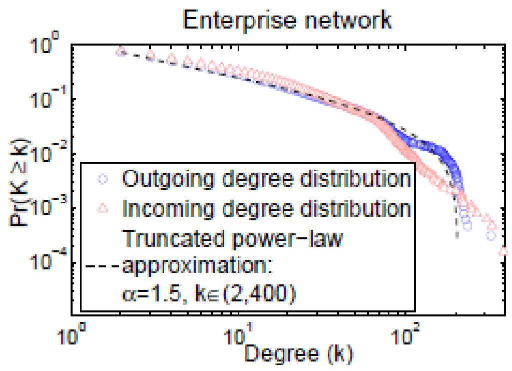
[Miao et al., WWW'12]

Network Model: Simulation results

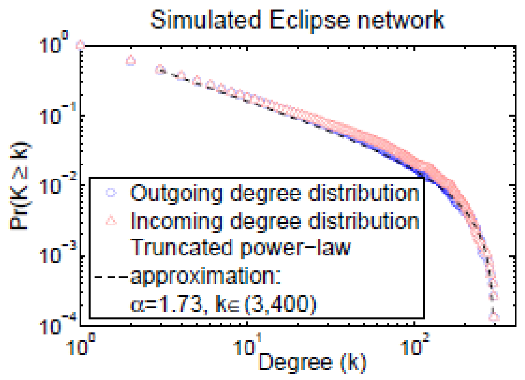
Real

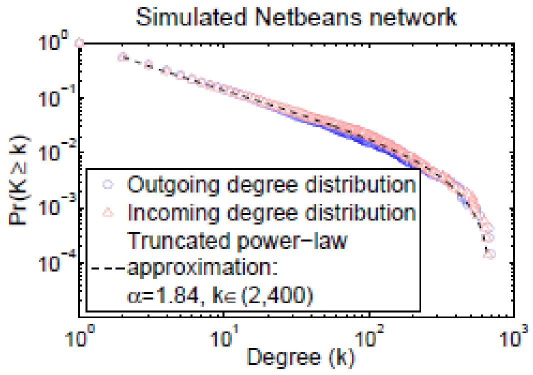


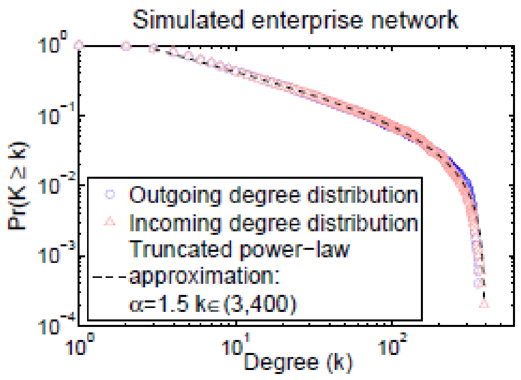




Simulation







UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

[Miao et al., WWW'12]

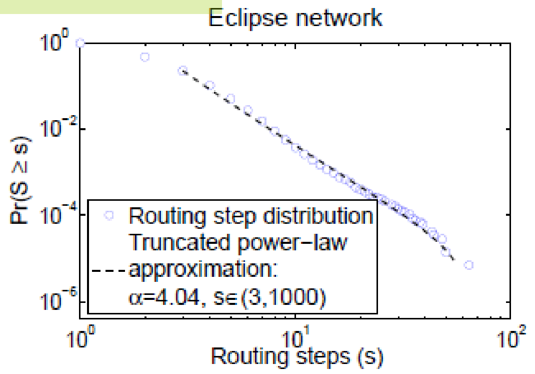
Routing Model: Simulating Task Execution

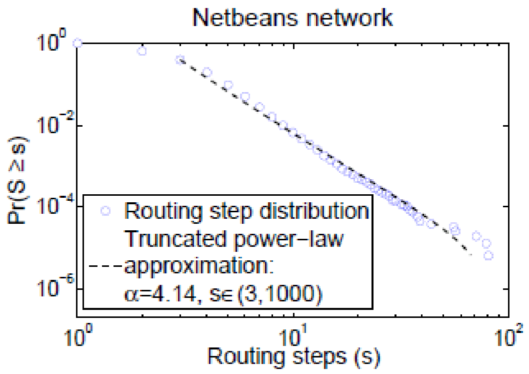
- Intuitions
  - If an expert cannot resolve a problem, she tends to forward it to a neighbor "closer to resolver"
  - If an expert doesn't know which neighbor is closer to resolver, she tends to choose a well-connected neighbor who is more likely to be connected to the resolver
  - Among all neighbors, an expert randomly chooses one to forward the task based on some probability distribution

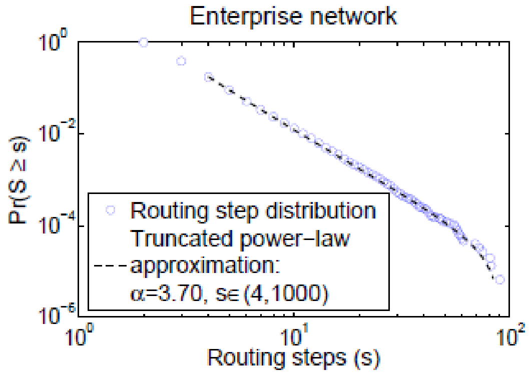
[Miao et al., WWW'12]

Routing Model: Simulation Results

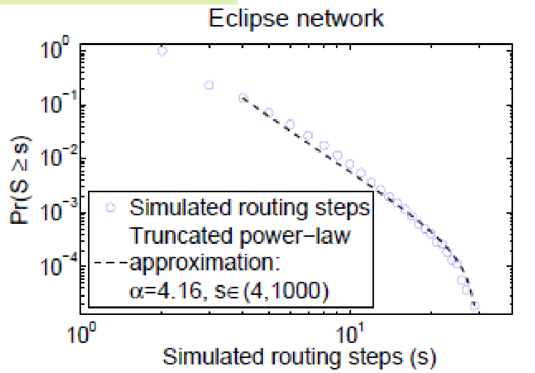
Real

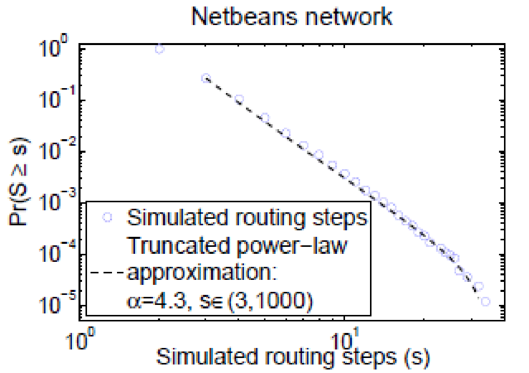


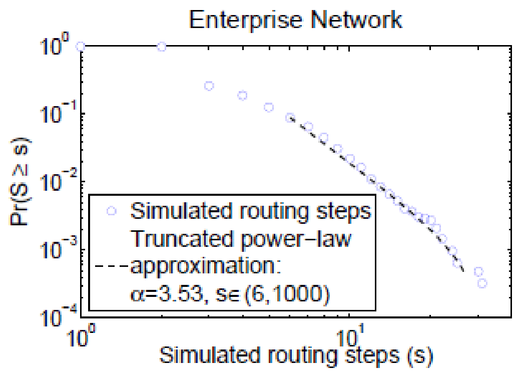




Simulation





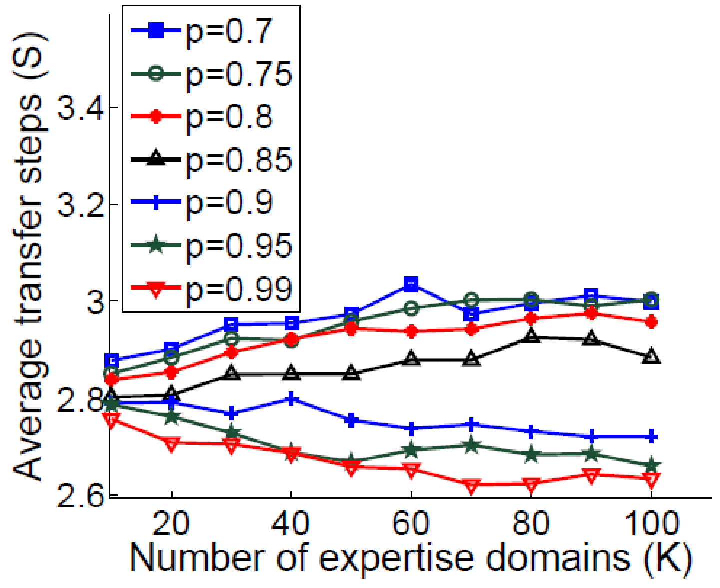


Note: Simulation results were obtained by running proposed routing model on a 2-D network topology, which is a neighborhood-preserving representation of real collaborative networks, using the method of spectral embedding.

[Miao et al., WWW’12]

Agent pool optimization: Putting the models in use

- **Environment**
  - A problem management function of an IT service provider, consisting of 5000 agents with varying expertise
- **Business Problem**
  - Constantly restructure agent pools to accommodate the evolving workload and human resource availability, while maintaining efficiency in problem resolution
- **Trade-offs**
  - Dividing agents into smaller pools
  - Dividing agents into larger pools
  - Correct initial dispatching (with probability *p*) is also critical to improving routing efficiency



[Heidari et al, AAAI'13]

How to effectively organize workforces?

❑Problem Definition

- Each task is associated with one difficulty value ***x***
- Each worker ***i*** is characterized with their ability ***w_i***
- Each worker can solve the task if and only if ***x<=w_i***
- If a worker cannot solve a given task, the task must be forwarded to a worker of greater ability.

Goal:

Given a set of worker abilities ***W*** and a distribution ***P*** over task difficulty, design efficient forwarding structure, that minimizes both <u>the maximum workload of any worker</u>, and <u>the number of workers that need to attempt a task</u>.

[Heidari et al, AAAI’13]

How to effectively organize workforces?

- Results
  - Omniscient workload
    - Assign a task to any worker that is capable of solving it.
    - For any ***W*** and task difficulty distribution ***P***, precisely characterize the optimal omniscient maximum workload ***M***
  - Near-Optimal depth-workload tradeoffs
    - By hierarchically arranging the workers in balanced ***b***-ary trees, we can simultaneously obtain a multiplicative factor of **b^2** of ***M*** in terms of maximum workload, with a resulting depth of ***log*(*n*)/*log*(*b*).**

Collaborations In This Talk

- ❑ **Peer Interaction**
  - o Patterns
  - o Prediction
  - o Optimization

- ❑ **Team Joint Collaboration**
  - o Patterns
  - o Prediction
  - o Optimization

Collaborative Teams

- ❑ Work together to produce music, movies, games, and other cultural products.
  - o Wikipedia
  - o Open Source Software (OSS)

Questions:

- ❑ How do teams form?

- ❑ What factors will influence the success of teams?

[Settles et al., CHI’13]

The Formation and Success of Online Creative Collaborations

- ❑ Existing theories on common bond, social identity, and social exchange [Kraut et al., 2007, Taifel et al, 1982, Emerson et al., 1976]
  - o Communication richness
  - o Shared interests
  - o Status within the community
  - o Balance of efforts

[Settles et al., CHI’13]

The Formation and Success of Online Creative Collaborations

- Data
  - February Album Writing Month (FAWM)
    6116 users; 39,103 FAWM songs

- Goal: to predict whether a pair of users posted a collaborative song to the website or not.

- Method
  - Logistic regression

[Settles et al., CHI'13]

The Formation and Success of Online Creative Collaborations

- Features:
  Various Paths that can connect two users in FAWM network
  e.g.,
  A-->follows--B,
  A<--messaged--B

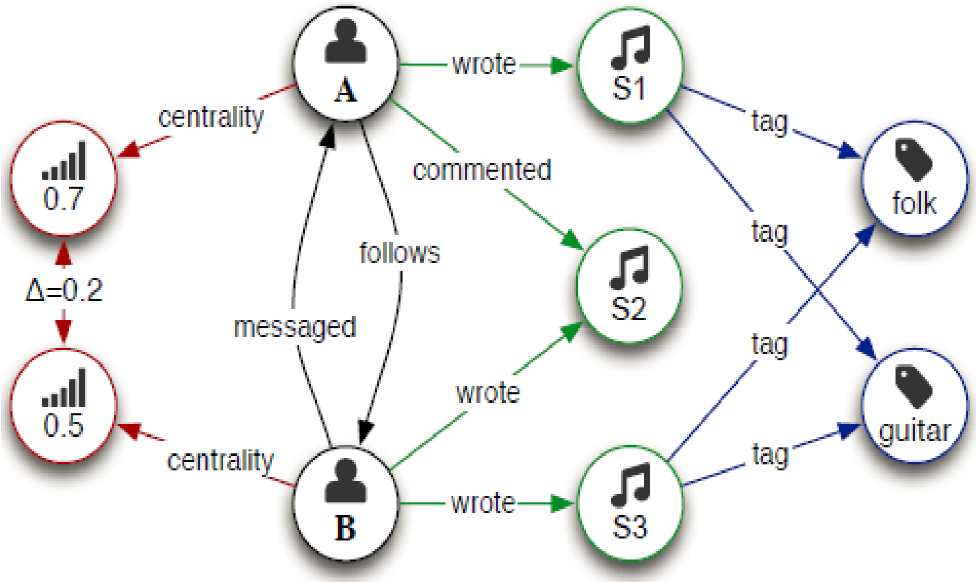
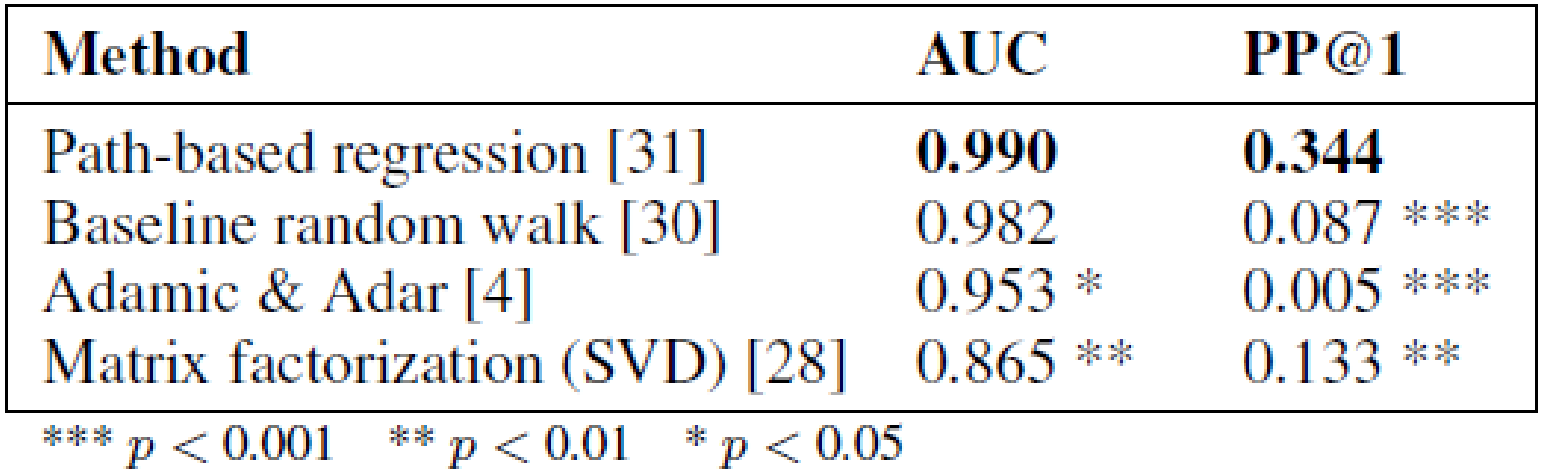


Figure 1. A small example subset of the FAWM network.

[Settles et al., CHI'13]

The Formation and Success of Online Creative Collaborations

❑Results



Method	AUC	PP@1
Path-based regression [31]	**0.990**	**0.344**
Baseline random walk [30]	0.982	0.087 ***
Adamic & Adar [4]	0.953 *	0.005 ***
Matrix factorization (SVD) [28]	0.865 **	0.133 **

*** $p < 0.001$ ** $p < 0.01$ * $p < 0.05$

Table 1. Evaluation of link-prediction methods, in terms of area under the ROC curve (AUC) and personalized precision at rank one (PP@1).

[Settles et al., CHI'13]

The Formation and Success of Online Creative Collaborations

❑ Results

Key findings on team success:

1. Balance of effort improves satisfaction.

2. Higher-status partners may enjoy Collabs less.

3. Frequent communication helps (usually).

[Settles et al., CHI'13]

The Formation and Success of Online Creative Collaborations

❑Results

Key findings on team formation:

1. Communication Exchanges predict collaboration, such as following a partner's song feed, direct messaging, and commenting on the partner's songs.

2. Collabs form out of shared interests, but different skills

3. Small status differences are positively associated with collabs. (Neither too different, nor exactly the same. similar to our finding in KDD'14)

[Luther et al., Group'10]

Success Factors in Online Creative Collaboration

- Collaborative animated movies (Collab)
  - Newgrounds: 2,200,000 registered users; 180,000 animated movies and games

- Success definition:
  - The completion of a Collab

[Luther et al., Group’10]

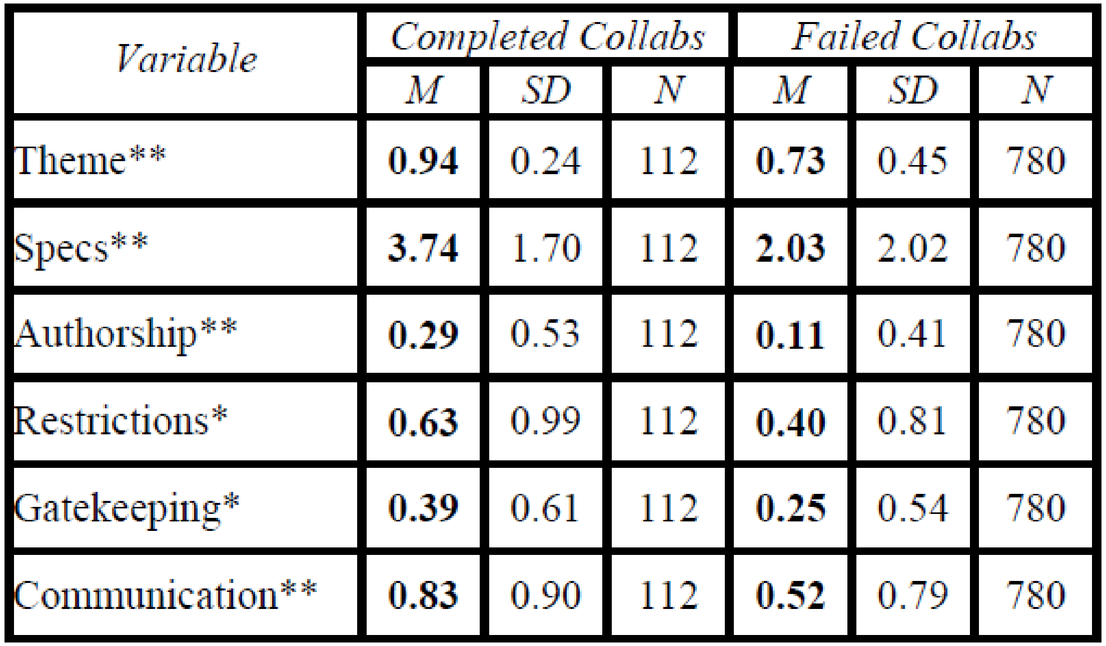
Success Factors in Online Creative Collaboration

- Three factor types:
  - Planning & Structure
    - Collabs with initial planning and structure, especially technical specifications, are more likely to be successful.
  - Reputation & Experience
    - Animators who are well-known in the community are more likely to lead successful collabs.
    - Animators who have experience with Flash and past collabs are more likely to lead successful collabs.
  - Communication & Dedication
    - Collabs whose members frequently communicate are more likely to be successful
    - Collabs whose leaders frequently communicate are more likely to be successful.

[Luther et al., Group'10]

Success Factors in Online Creative Collaboration

- Experiments



Variable	*Completed Collabs*			*Failed Collabs*		
	M	*SD*	*N*	*M*	*SD*	*N*
Theme**	**0.94**	0.24	112	**0.73**	0.45	780
Specs**	**3.74**	1.70	112	**2.03**	2.02	780
Authorship**	**0.29**	0.53	112	**0.11**	0.41	780
Restrictions*	**0.63**	0.99	112	**0.40**	0.81	780
Gatekeeping*	**0.39**	0.61	112	**0.25**	0.54	780
Communication**	**0.83**	0.90	112	**0.52**	0.79	780

Table 1. Comparison of first post attributes for completed vs. failed collabs, * $p < 0.05$, ** $p < 0.01$.

Each row stands for one category of planning/structural elements in the first posts of Collab threads. Completed Collabs tend to have more planning and structure than failed ones.

Part III: Sum-Up

Collaborations:

❑Peer Interaction

❑Team Joint Collaboration

Questions:

- How do experts make routing decisions?
- Who have made inefficient routing decisions?
- How to optimize the routing performance through targeted training?
- Can the completion time of a task be predicted so that one can act early for difficult tasks?
- Can we predict next best experts?
- How to model the collaborative networks?

- How do teams form?
- What factors will influence the success of teams?

References

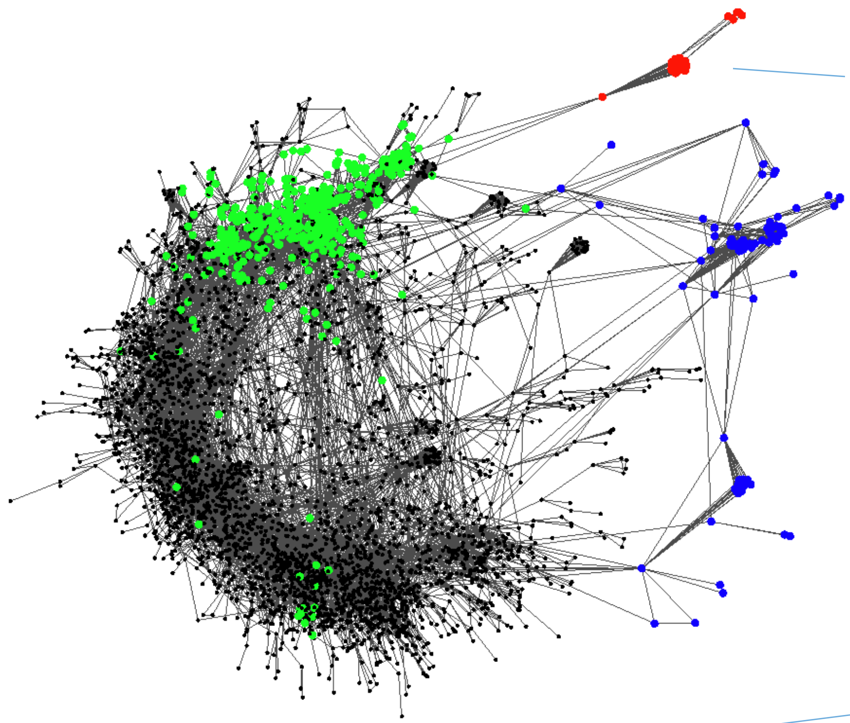
- Analyzing Expert Behaviors in Collaborative Networks. [Sun et al., KDD’14]
- Efficient Ticket Routing by Resolution Sequence Mining. [Shao et al., KDD’08]
- Generative Models for Ticket Resolution in Expert Networks. [Miao et al., KDD’10]
- Understanding Task-Driven Information Flow in Collaborative Networks. [Miao et al., WWW’12]
- Depth-Workload Tradeoffs for Workforce Organization. [Heidari et al., AAMAS’12]
- Why It Works (When It Works): Success Factors in Online Creative Collaboration. [Luther et al., GROUP’10]
- The Formation and Success of Online Creative Collaborations [Settles et al., CHI’13]

Part IV
Relationship Mining

Relationships embedded in network structure

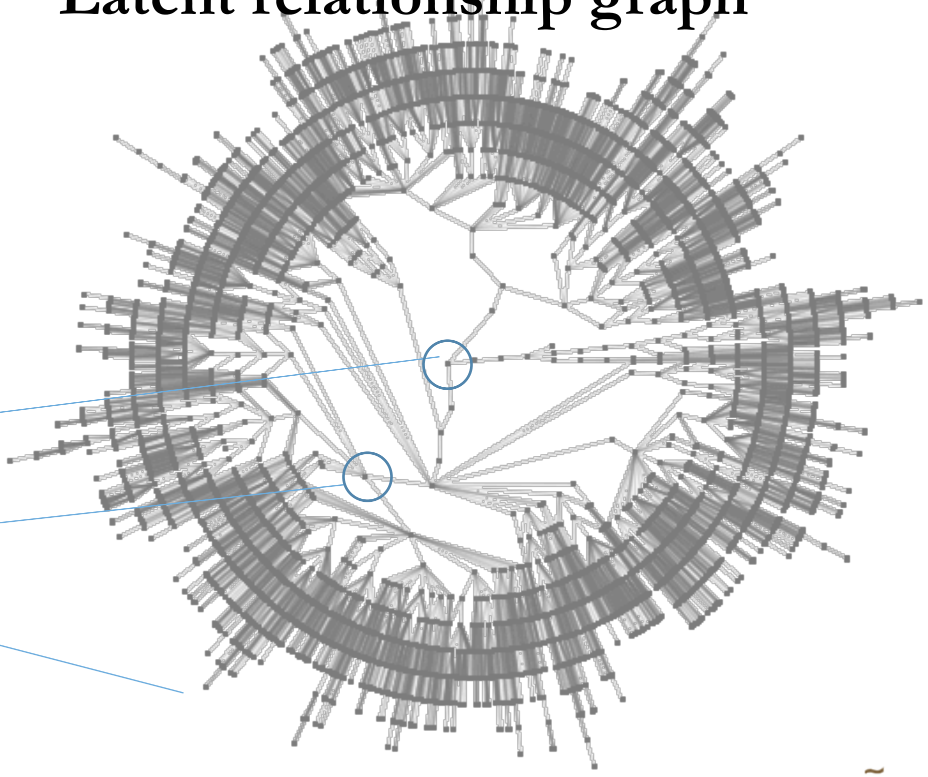
- Network links represent general relationships of longer term
  - Friendship of mutual consent, e.g., Facebook.
  - Follow link, e.g., Twitter.
  - Collaboration, e.g., DBLP.
  - Negative relationships, e.g., distrust (Epinion), foe (slashdot).
- Can we discover more specific and implicit relationships from these links?

Example: Role Discovery



Information network without role/relationship info, e.g. a company's email network

Latent relationship graph



How to infer

- **CEO**
- **Manager**
- **Employee**

Courtesy of Chi Wang for this slide

Twitter is a unique social platform

1. The "follow" links are established without mutual consent.
   - An explosion of social links
     - Everyone has a large number of followers and followees.
     - A huge number of tweets are generated everyday. (~300 million tweets/daily)
   - A further shrinkage of the network diameter
     - Information diffusion is much faster
2. It is a mixture of social network and news media
   - H.Kwak et al WWW 2010

Follow network = real-life social network?

How much of this follow network reflects a user's real-life offline social network?

- Mutual follow links do not necessarily indicate real-life interaction.
- The number of followees and followers varies significantly.

Problem: Given a Twitter follow network of a target user, identify the user's offline community by examining the follow linkage alone.

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

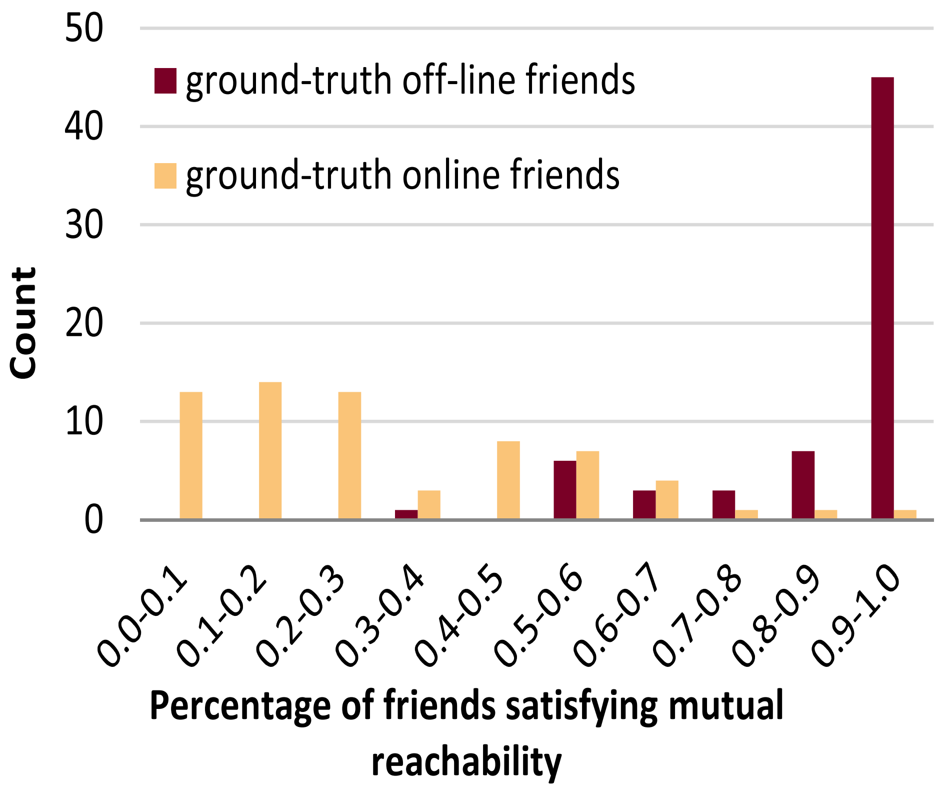
Motivation

1. More accurate and robust user interest modeling
2. Online vs Offline relationship understanding
3. User identity alignment across different platforms
4. Spam, Zombie account detection
5. Business competitive analysis

And more …

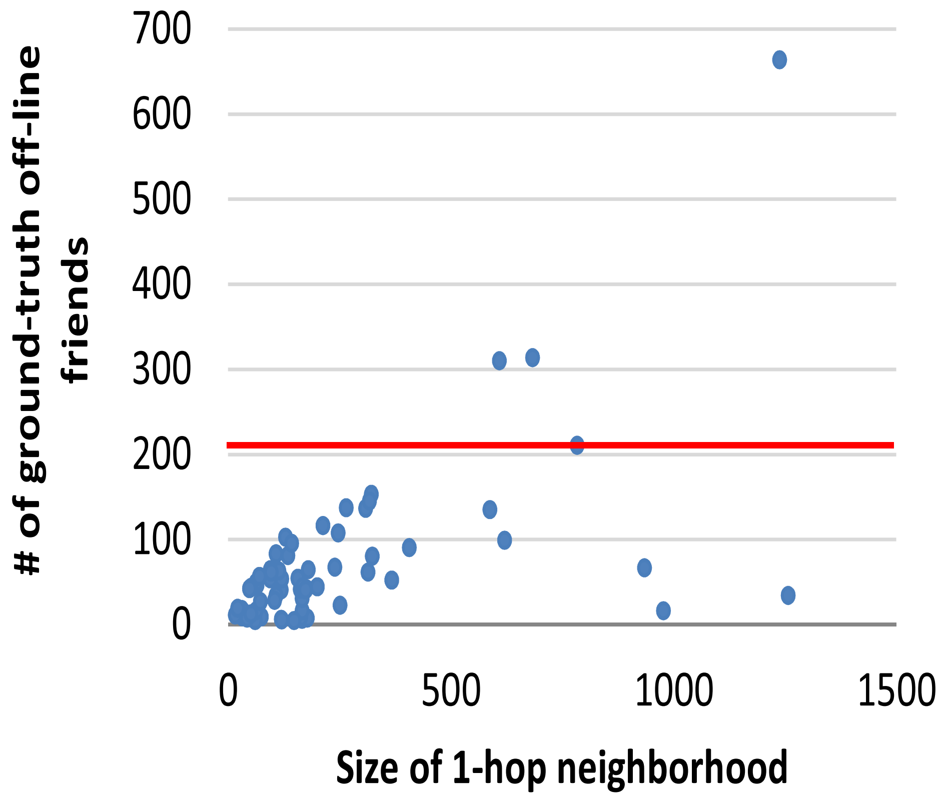
Principle I: Mutual Reachability

Information should be able to flow in both directions within a small distance between real-life friends.



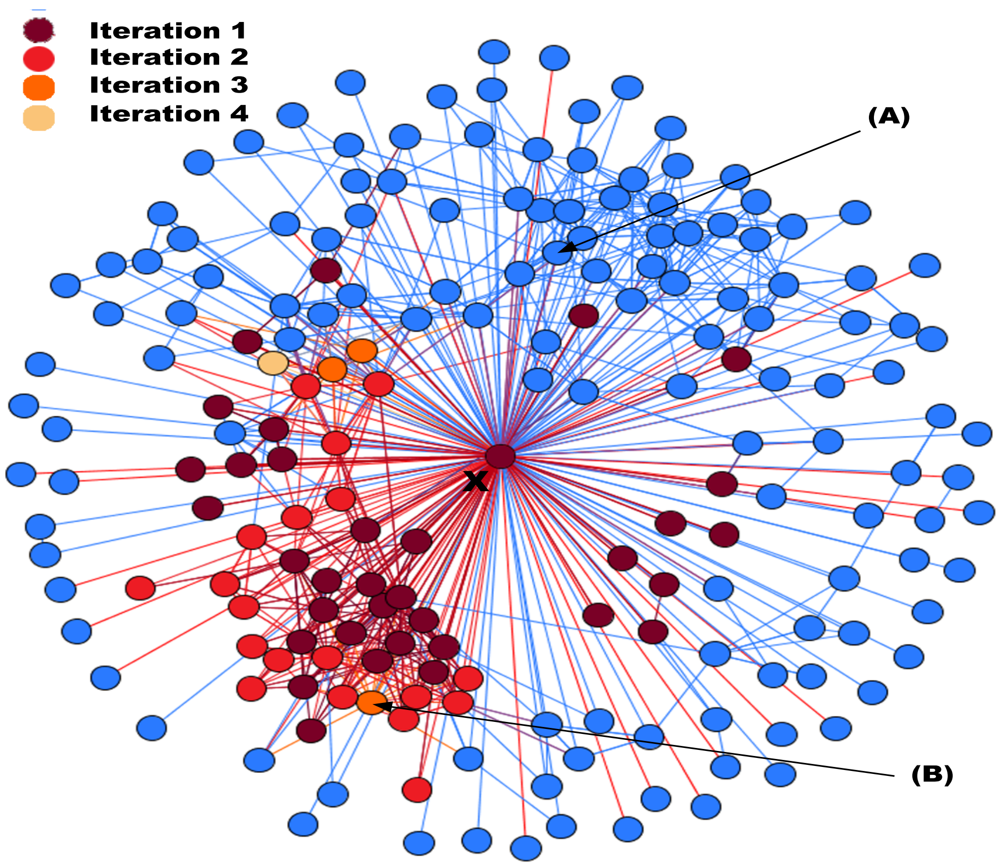
Principle II: Friendship Retainability

The size of a user's offline community has an upper-bound threshold σ related to Dunbar's number



Principle III: Community Affinity

A user's off-line friends usually group into clusters such that within each cluster members know each other



Approach

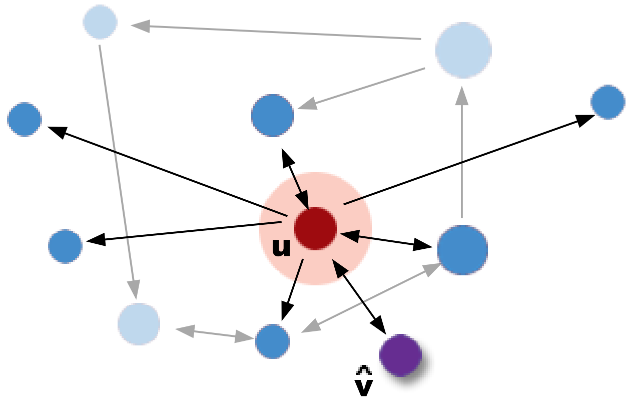
- Random Walk with Restart

$$\vec{r_i} = (1-c)\tilde{W}\vec{r_i} + c\vec{e_i}$$

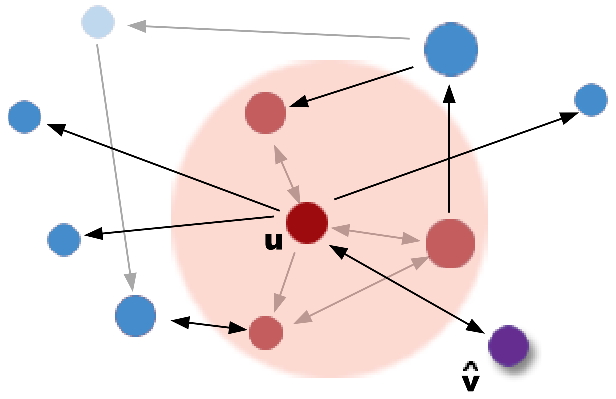
- Closeness Score

$$c_{i,j} = r_{i,j} * r_{j,i}$$

- Iterative Off-line Community Discovery
  - Off-line community is discovered by iterations.
  - A virtual user node is used as the threshold to cut for each iteration.



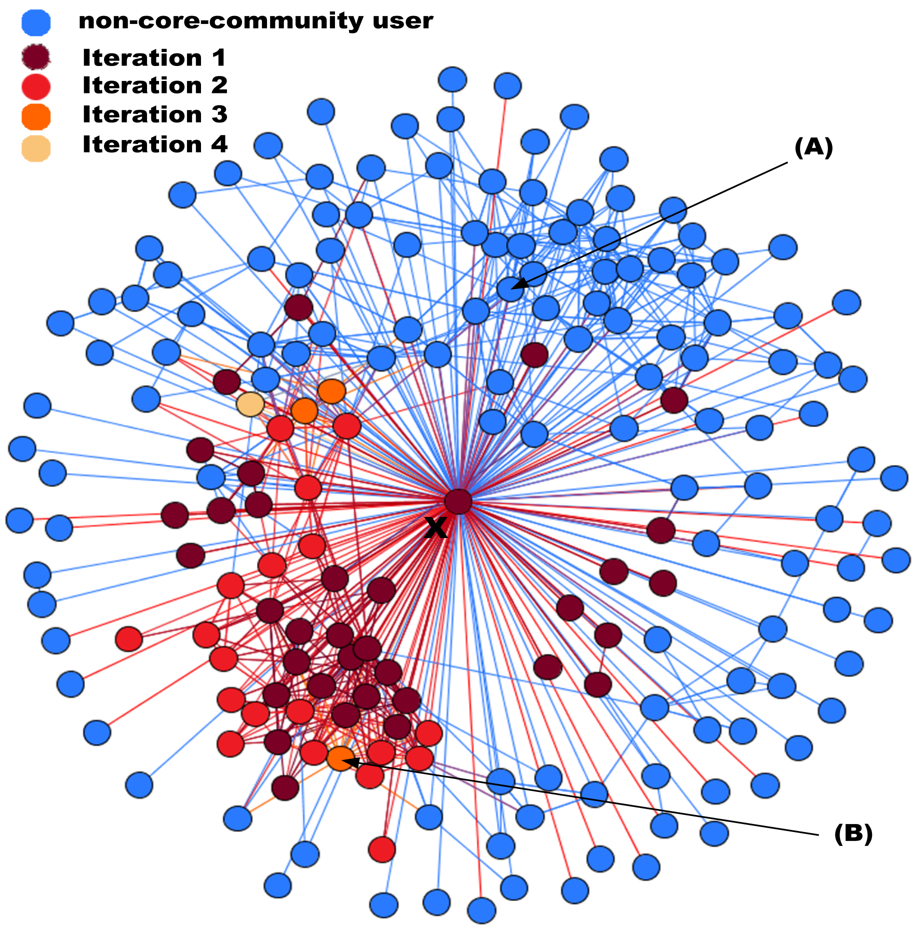
Iteration 1

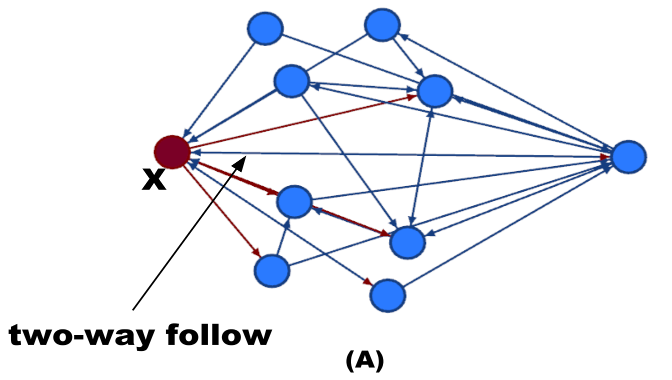


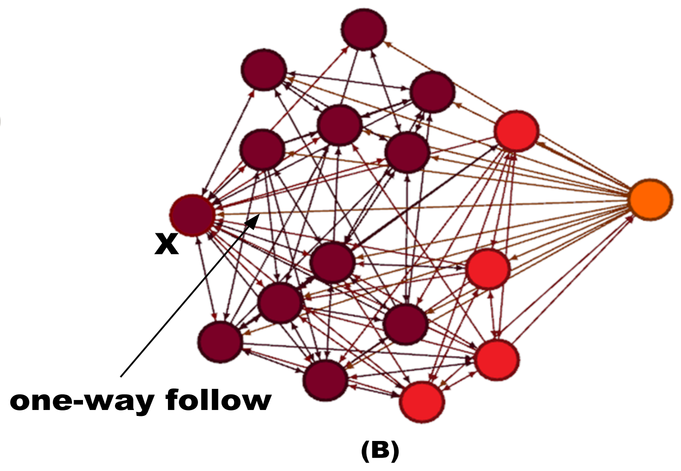
Iteration 2

A Case Study

A real Twitter user: following 385 users; followed by 107 users

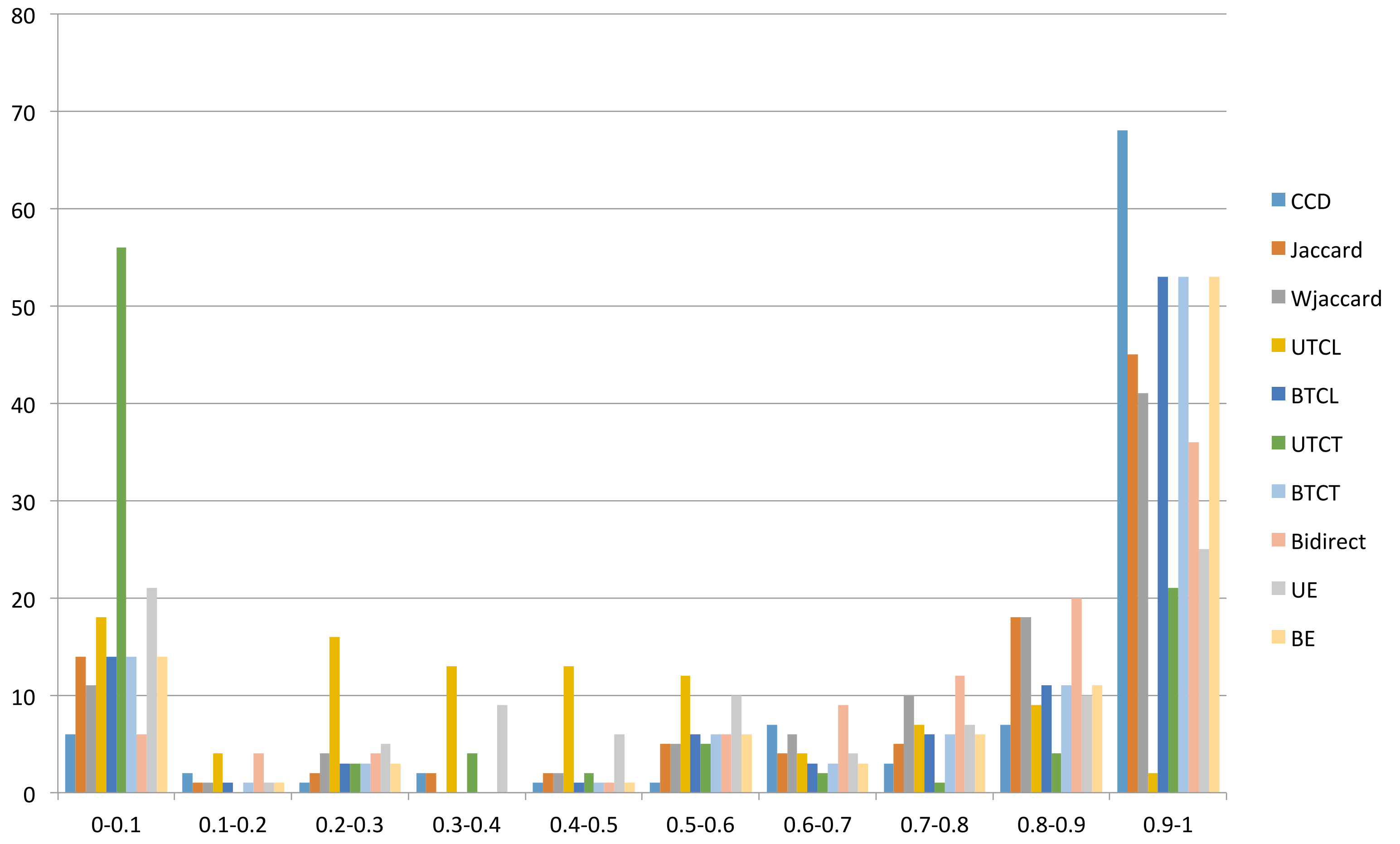






Precision

Our method, Core Community Discovery (CCD), performs the best.

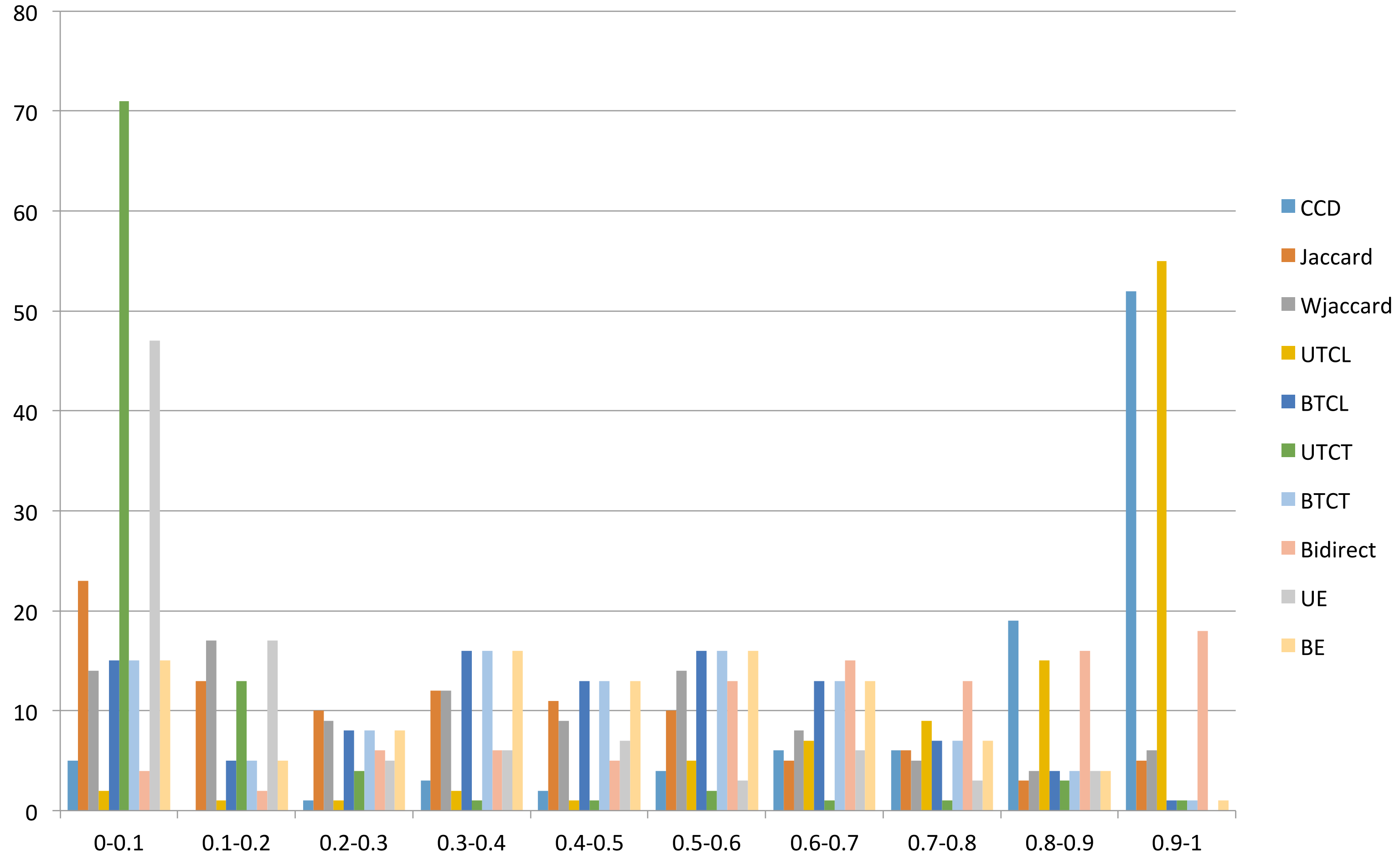


UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

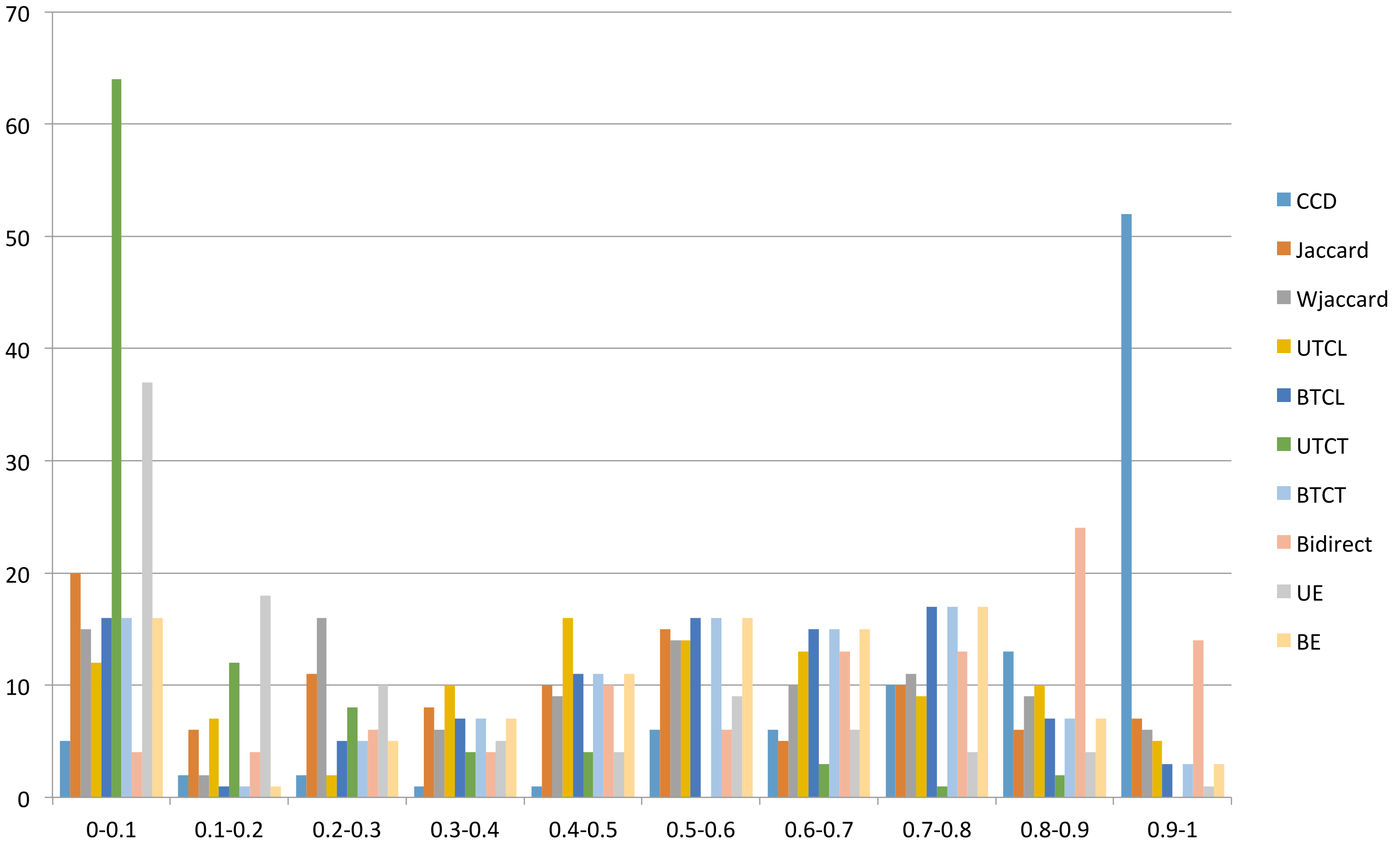
Recall

Our method, Core Community Discovery (CCD), performs the best.



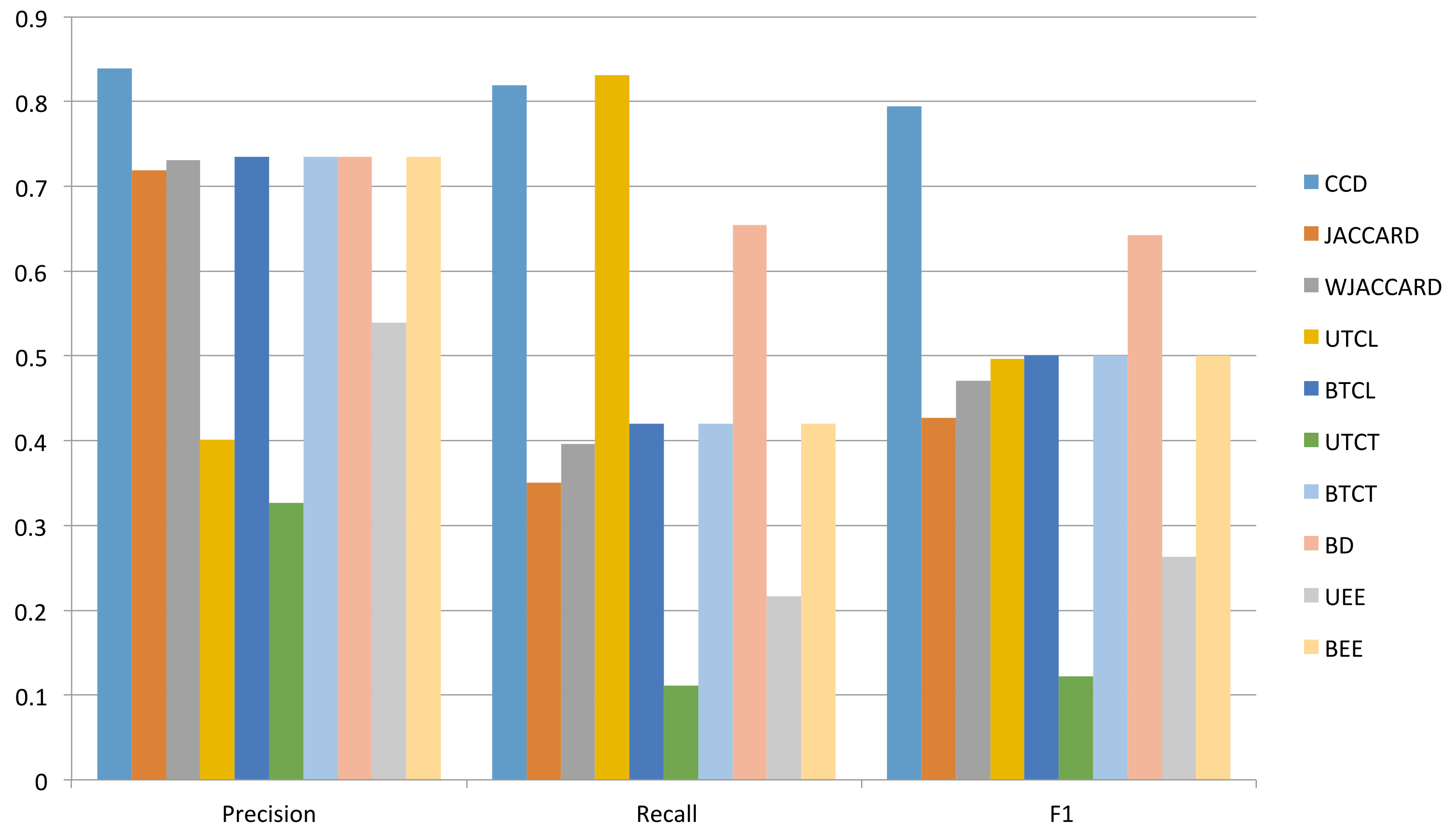
F-score

Our method, Core Community Discovery (CCD), performs the best.



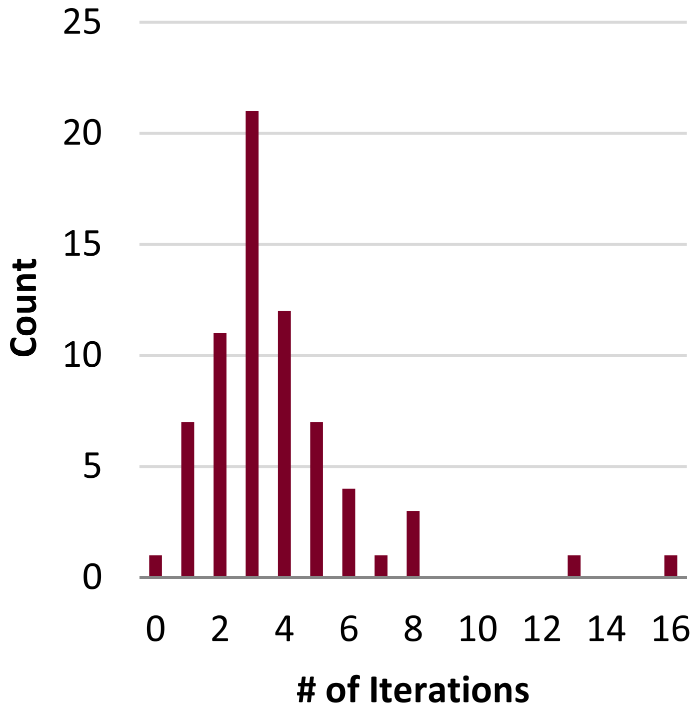
ALL

Our method, Core Community Discovery (CCD), performs the best.

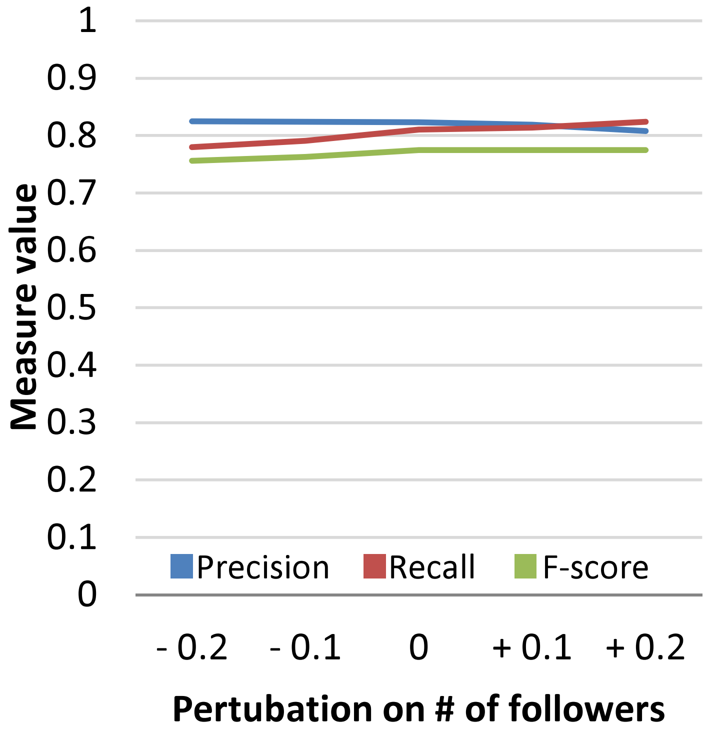


Parameters

- **On # of Iterations**

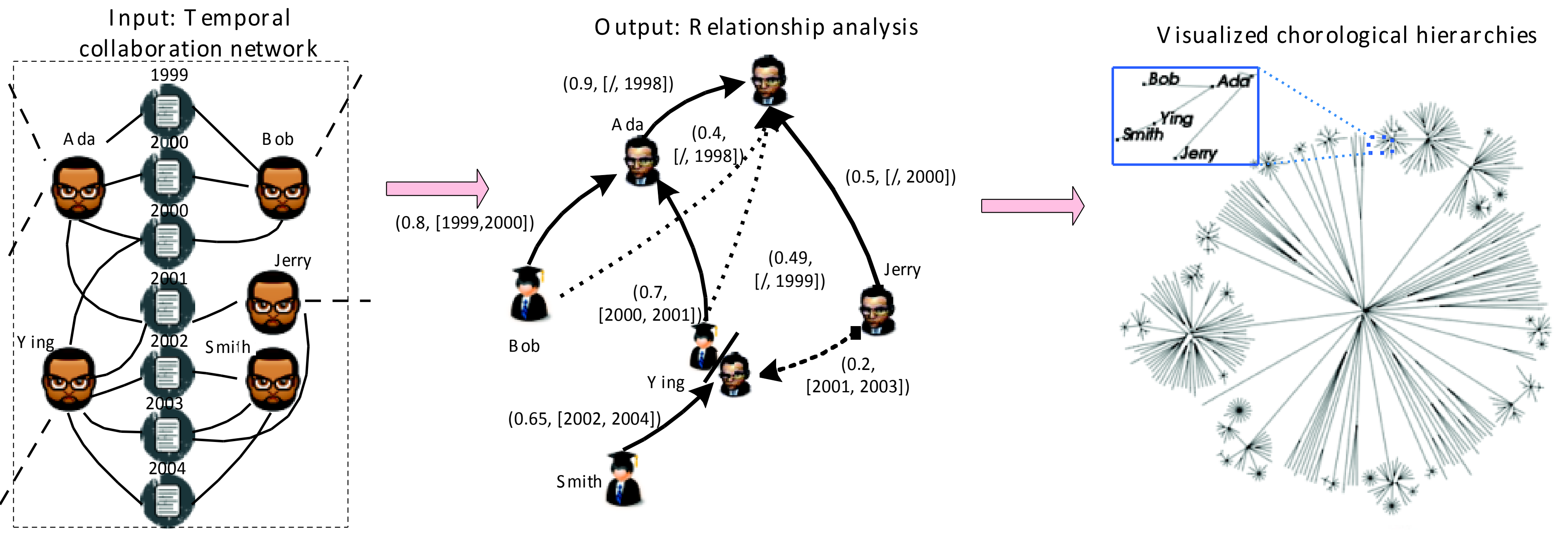


- **On Robustness**



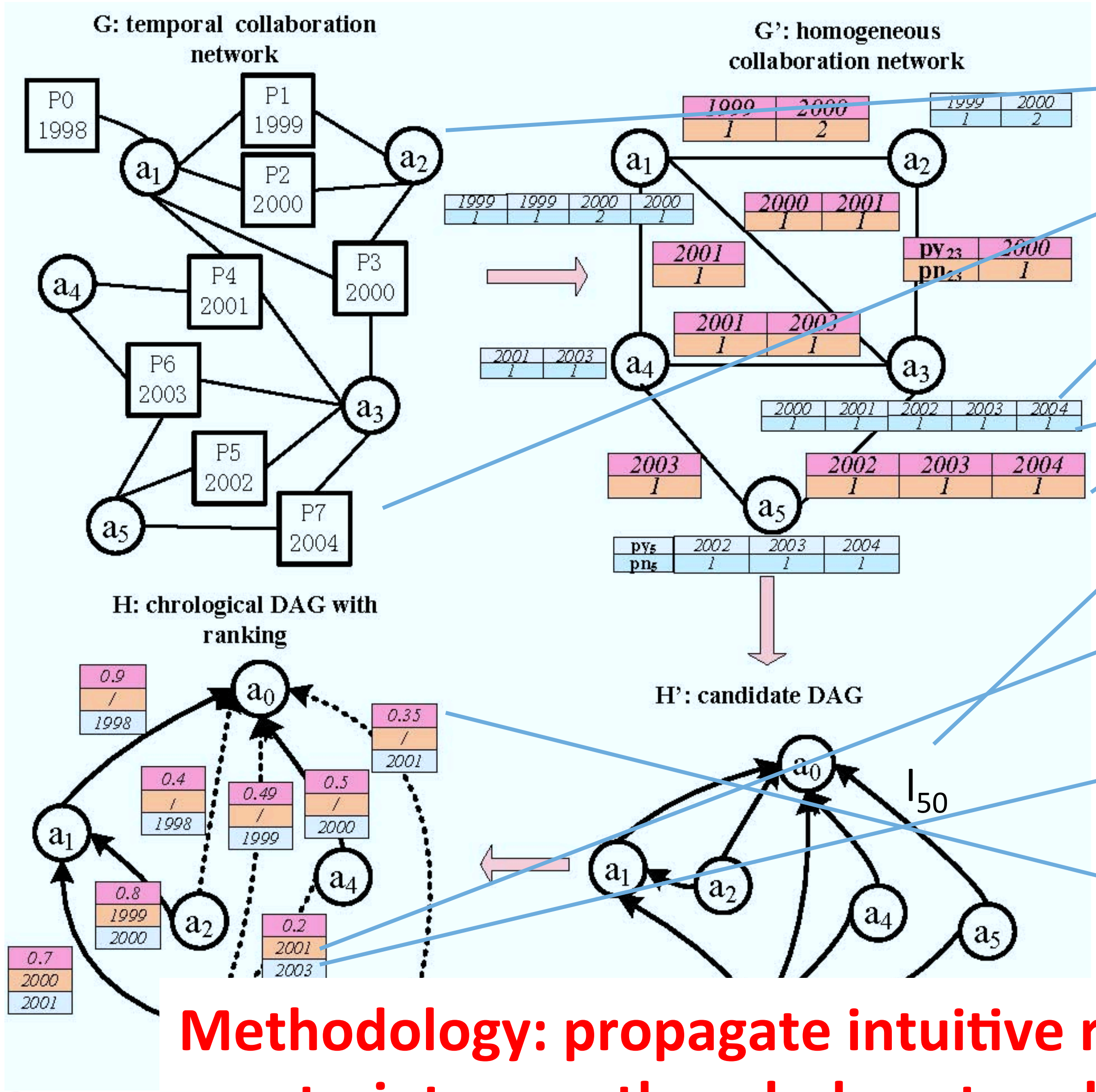
Mining Advisor-advisee Relationship [Wang et al, KDD'10]

- Input: research publication network.
- Output: potential advising relationship and their ranking – (r, [st, ed])



Courtesy of Chi Wang for this slide

Overall Framework



G: temporal collaboration network
G': homogeneous collaboration network
H: chrological DAG with ranking
H': candidate DAG
l_{50}

- a_i: author i
- p_j: paper j
- py: paper year
- pn: paper#
- l_{ij}: local feature
- $st_{i,yi}$: start time
- $ed_{i,yi}$: end time
- $r_{i,yi}$: ranking score

Methodology: propagate intuitive rules and constraints over the whole network.

Courtesy of Chi Wang for this slide

Basic Constraints

- If a_y advises a_x since the year st_x
  - a_y can only advise a_x after it graduated

  ➔ $ed_y < st_x < ed_x$
  - a_y must have a longer history of publication than a_x before st_x.

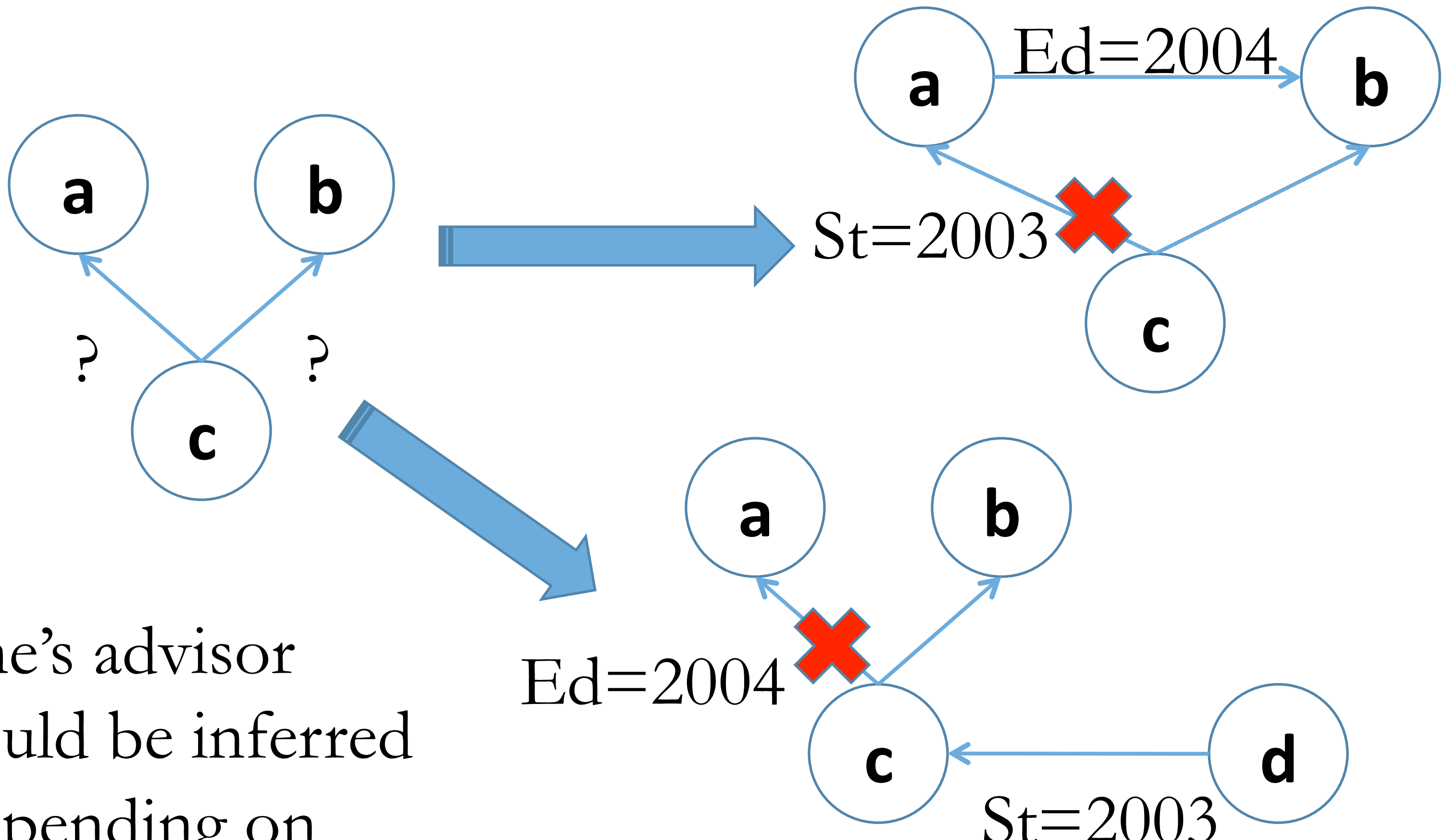
  ➔ The candidate graph H' is a DAG.

The model can incorporate other intuitions as factor functions into a Time-constrained Probabilistic Factor Graph (TPFG)

Courtesy of Chi Wang for this slide

Why is network structure helpful?

- More than pairwise features: interdependency



one's advisor could be inferred depending on others' advisor!

Courtesy of Chi Wang for this slide

Experiment on DBLP

- DBLP data: 654, 628 authors, 1076,946 publications, publishing time provided.
- Labeled data: MathGenealogy Project; AI Gealogy Project; Faculty Homepage

Datasets	RULE	SVM	TPFG	
TEST1	69.9%	73.4%	80.2%	**84.4%**
TEST2	69.8%	74.6%	81.5%	**84.3%**
TEST3	80.6%	86.7%	88.8%	**91.3%**

Heuristics

Supervised learning

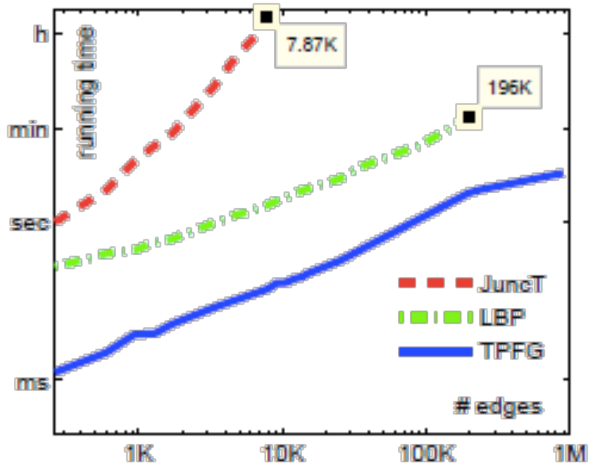
Empirical parameter

Optimized parameter

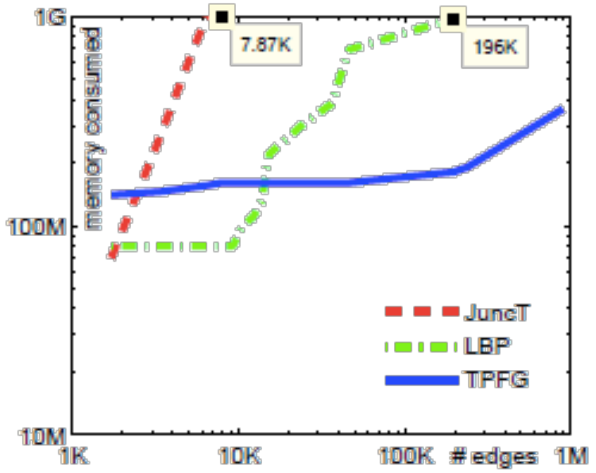
Courtesy of Chi Wang for this slide

Case Study & Scalability

Advisee	Top Ranked Advisor	Time	Note
David M. Blei	1. Michael I. Jordan	01-03	PhD advisor, 2004 grad
	2. John D. Lafferty	05-06	Postdoc, 2006
Hong Cheng	1. Qiang Yang	02-03	MS advisor, 2003
	2. Jiawei Han	04-08	PhD advisor, 2008
Sergey Brin	1. Rajeev Motawani	97-98	"Unofficial advisor"



(a) Time

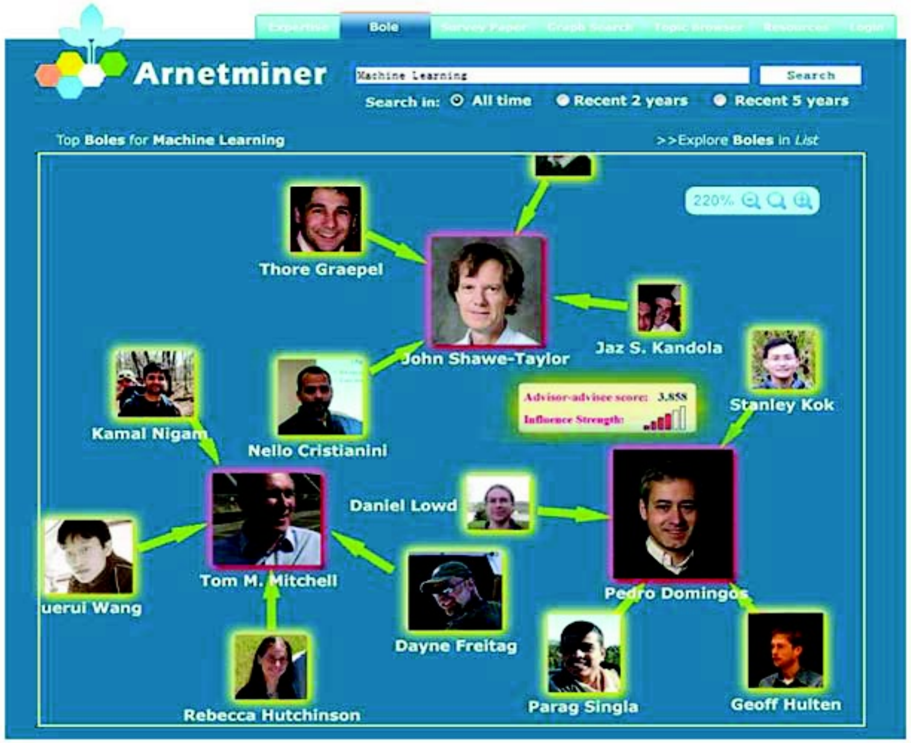


(b) Space

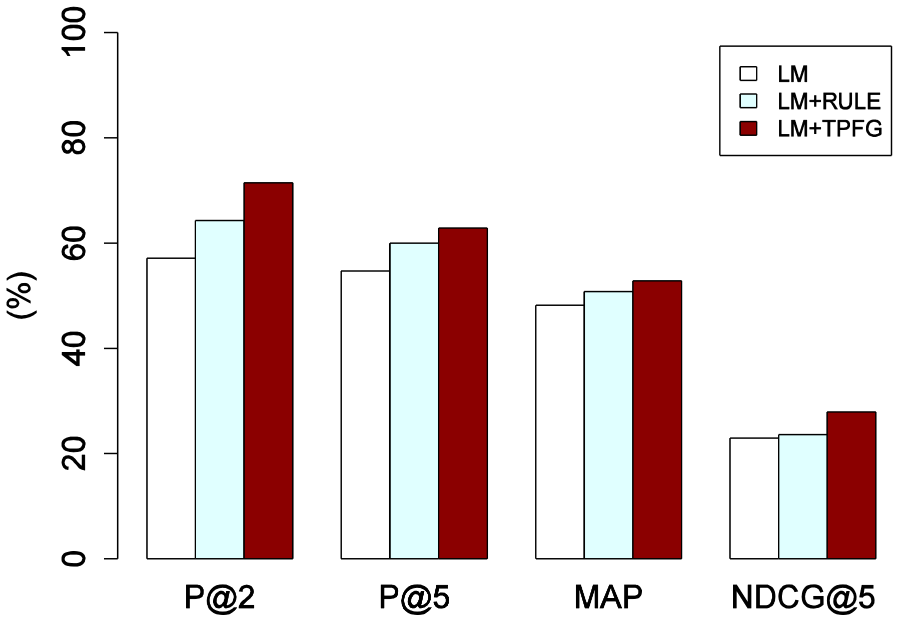
Courtesy of Chi Wang for this slide

Application: Expert Finding

An example on a real



Performance improvement



Courtesy of Chi Wang for this slide

Representative work

- "Relation Mining" [Kadri 03, Rinaldi 06, Coppola 08]
  - Mainly text mining and language processing on text data and structured data.
- "Relational Learning" [Getoor 07, Tang 09]
  - The classification when objects and entities are presented in multiple relations
- Relationship with semantic meaning
  - [Diehl 07]: a supervised approach
- Positive-Negative relationship
  - Leskovec et al. [WWW'10]
- Social circle in ego network
  - McAULEY et al. [TKDD'14]

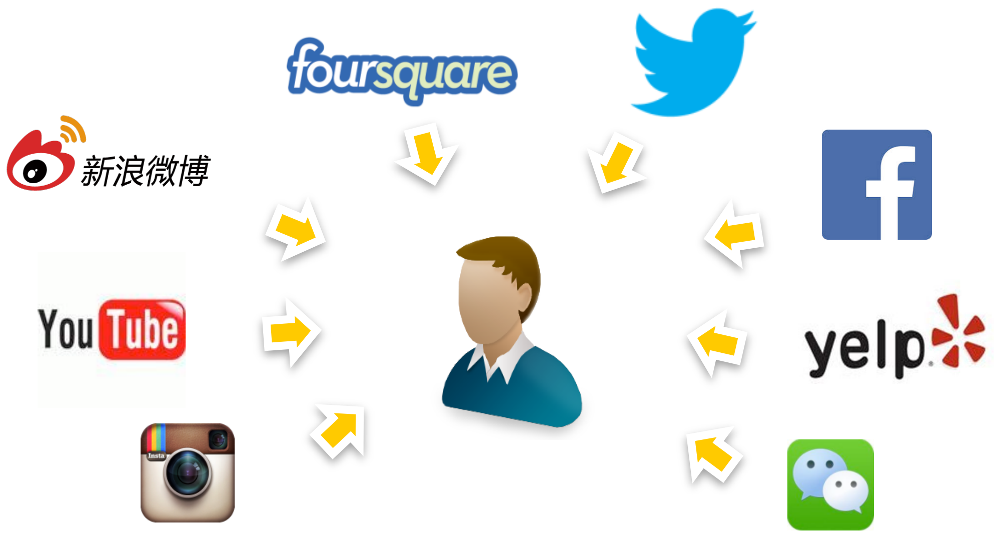
References

- W. Xie, C. Li, F. Zhu, E. Lim and X. Gong, ***"When a Friend In Twitter is a Friend In Life"***, ACM Int. Conf. on Web Science 2012.
- C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu and J. Guo , ***"Mining advisor-advisee relationships from research publication networks"***. KDD 2010.
- J. Leskovec, D. Huttenlocher and J. Kleinberg, "***Predicting Positive and Negative Links in Online Social Networks***", WWW 2010.
- J. Mcauley and J. Leskovec, "***Discovering social circles in ego networks***", TKDD, 2014.

Part V
User Identity Linkage

User Identity Linkage

Link up all the accounts of the **same** user across **different** social platforms

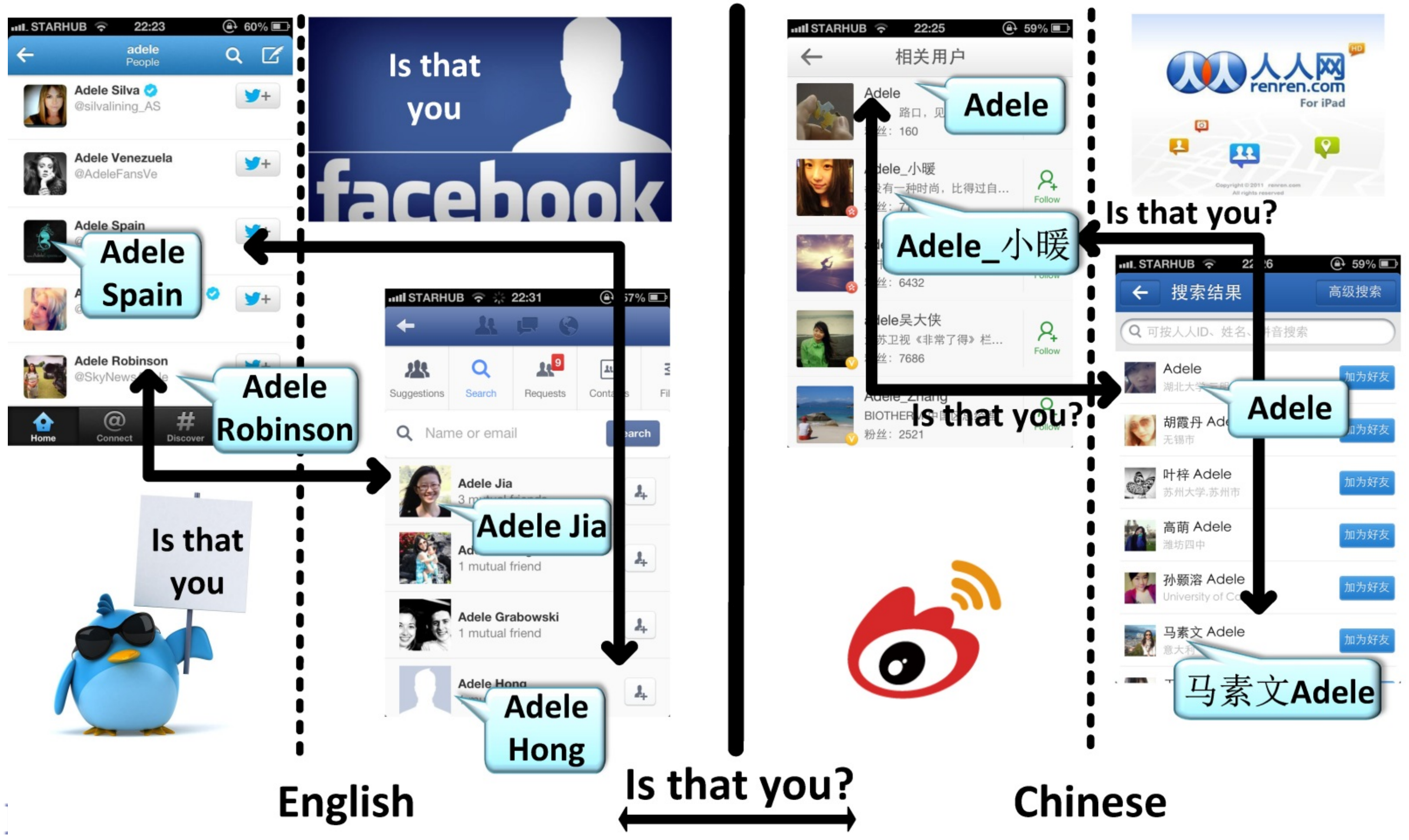


Why do we care?

- **Completeness**
  - Cross-platform user linkage would enrich an otherwise fragmented user profile to enable an all-around understanding of a user's interests and behavior patterns.
- **Consistency**
  - Cross-checking among multiple platforms helps improve the consistency of user information.
- **Continuity**
  - User identity linkage makes it possible to integrate useful user information from those platforms that have over time become less popular or even abandoned.

Example: A Not So Easy Case

Search name *Adele* in different social media platforms

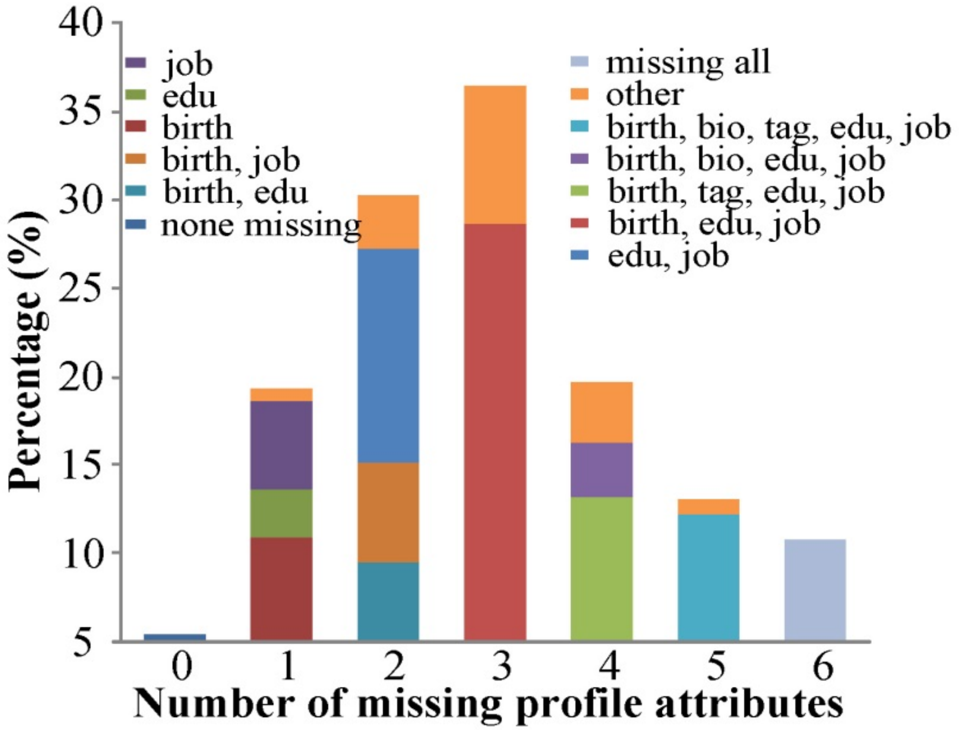


Research Challenges

- **Usernames are not always reliable**
  - Traditional approaches that heavily rely on username parsing to link users may fail on more diversified communities.
  - Statistical models (e.g. SVM) or rule based models constructed with mere username and attribute analysis are far from being robust to accurately identify user linkage across online social communities.

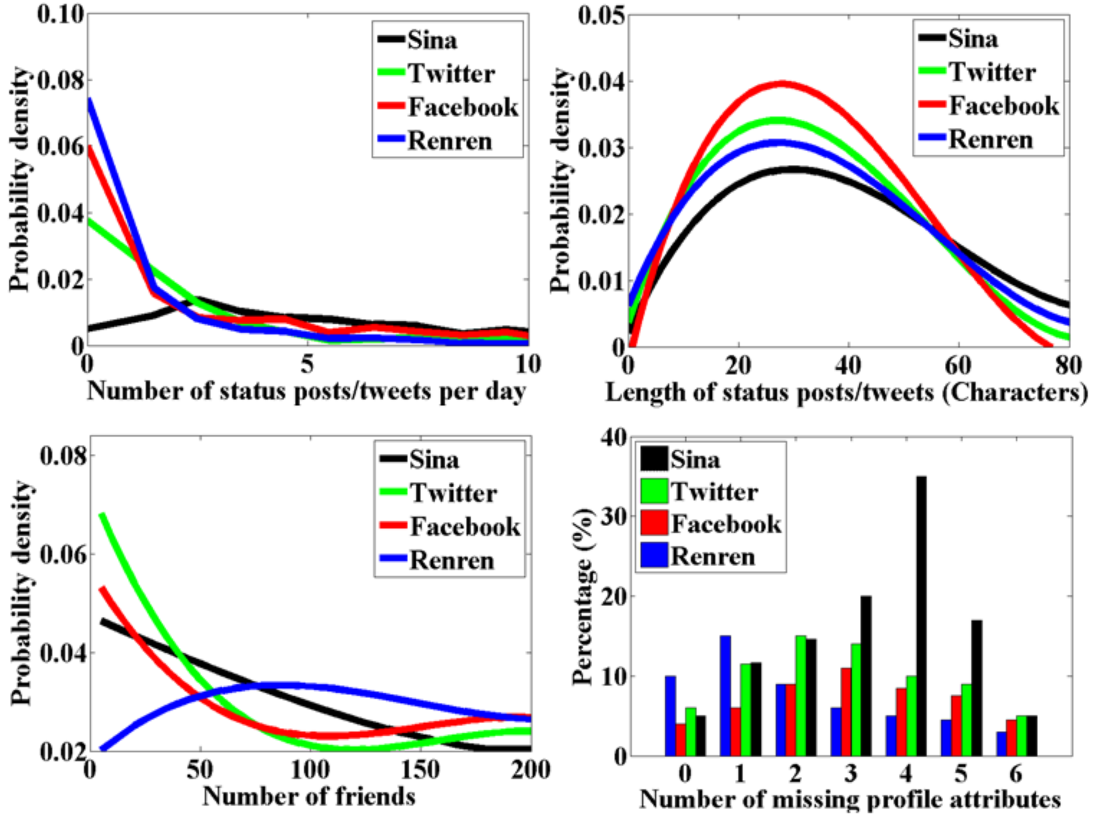
Research Challenges

- **Missing Information**
  - At least 80% of users are missing at least 2 profile attributes out of the 6 most popular ones, and merely 5% of users have all attributes filled up.



Research Challenges

- **Data Misalignment**
  - o Platform Difference.
  - o Heterogeneous behavior:
    - ➢ The user behavior can be represented by various types of media, e.g., locations, blogs, tweets, videos and images
  - o Behavior Asynchrony.
  - o Data Imbalance.



UCSB

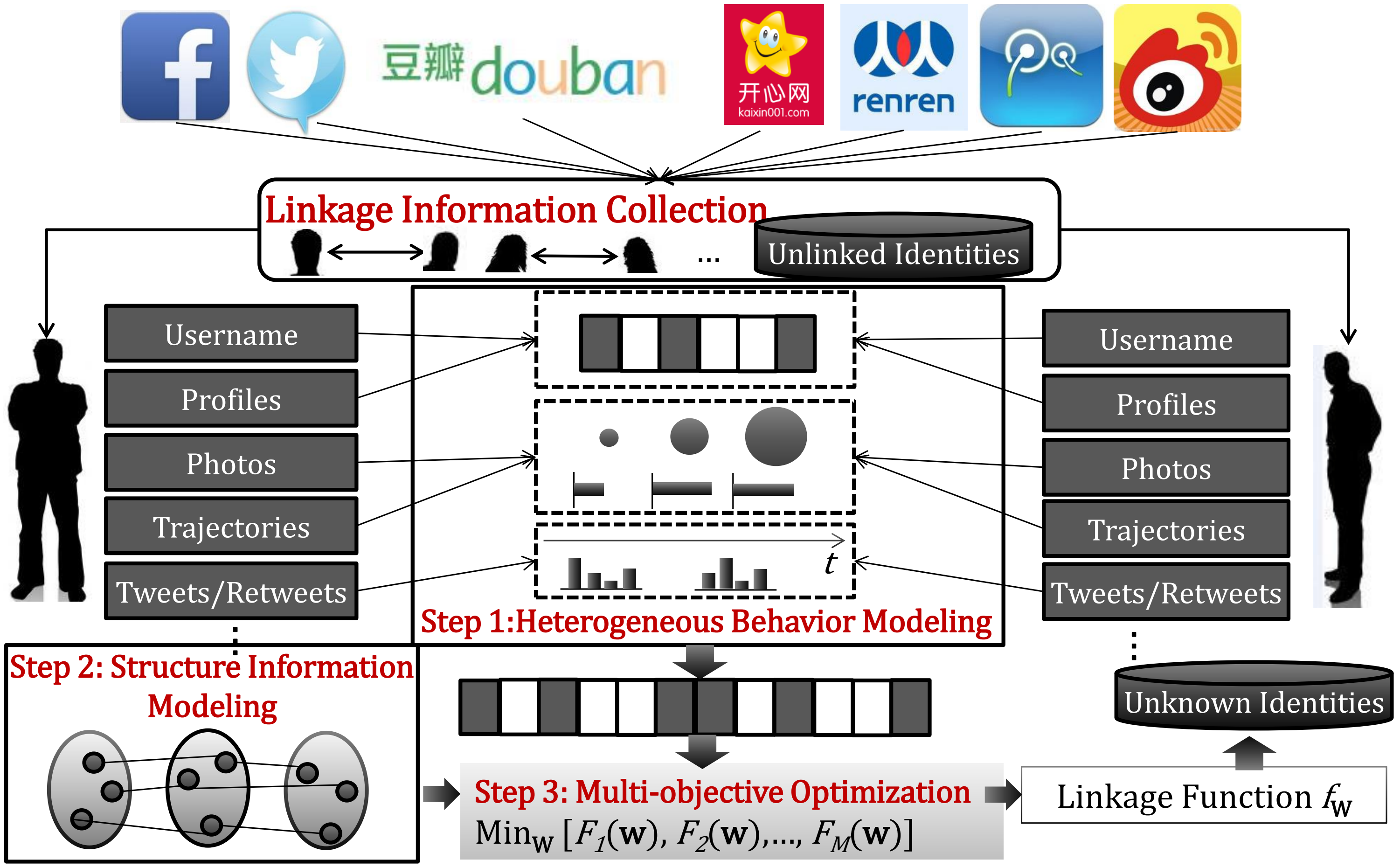
SMU SINGAPORE MANAGEMENT UNIVERSITY

Where are the hopes in social data?

Two important features unique to social data:

- User behavior trajectory along temporal dimension
  - Over a sufficiently long period of time, a user's social behavior exhibits a high level of consistency across different platforms.
- User's core social network structure
  - A user's core social network structures across different platforms share great similarity and offer a highly discriminative characterization of the user.

HYDRA: a user identity linkage framework [Liu et al. SIGMOD'14]



豆瓣 douban
开心网 kaixin001.com
renren
Linkage Information Collection
...
Unlinked Identities
Username
Profiles
Photos
Trajectories
Tweets/Retweets
t
Step 1:Heterogeneous Behavior Modeling
Step 2: Structure Information Modeling
Unknown Identities
Step 3: Multi-objective Optimization
Min$_w$ [F_1(w), F_2(w),..., F_M(w)]
Linkage Function f_w

UCSB

SMU SINGAPORE MANAGEMENT UNIVERSITY

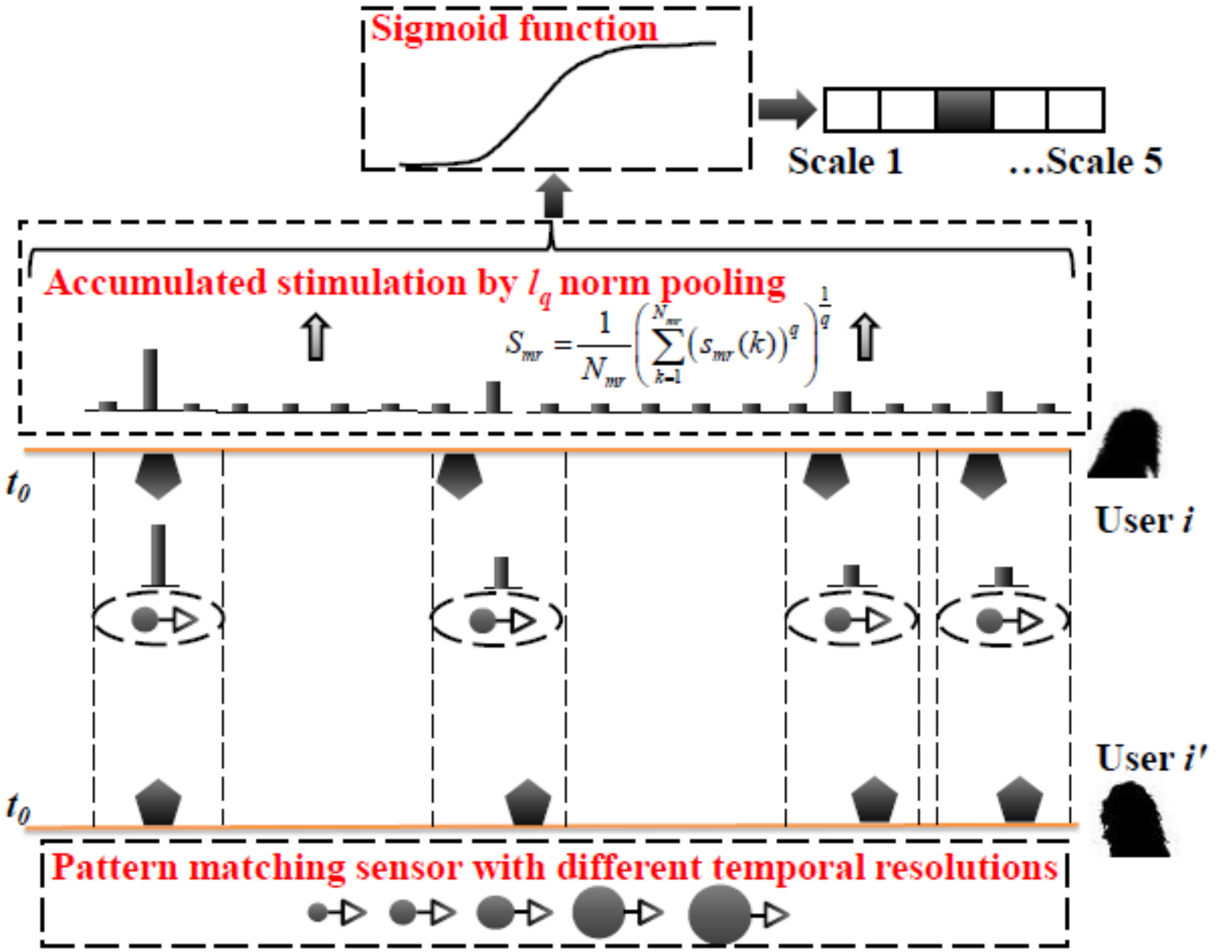
User Online Identity Data

- User attributes (numeric, categorical)
  - Demographics, location, personal interest, etc.
- User Generated Content (topics, sentiments)
  - Reviews, tweets, ratings, multimedia, etc.
- User Core Social network (snapshot/static view)
  - Friend network, followers/followees network, communities/interest groups, etc.
- User Behavior trajectory (dynamic, evolutionary)
  - content sharing history, social interaction pattern, network formation, etc.

Main Stages

- Behavior Similarity Modeling
  - Calculate the multi-dimensional similarity vector between two users of a pair for all user pairs via heterogeneous behavior modeling.
- Structure Information Modeling
  - Construct the structure consistency graph on user pairs by considering both the core network structure of the users and their behavior similarities.
- Multi-objective Optimization with Missing Information
  - A two-class classification model via optimizing two kinds of objective functions simultaneously.

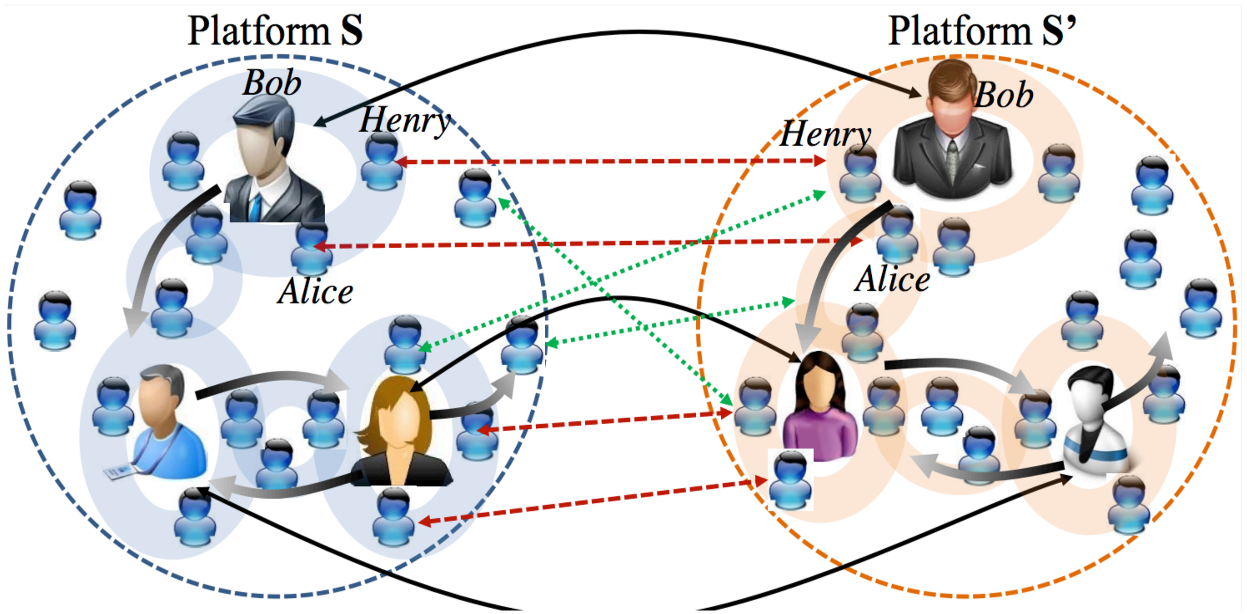
Multi-resolution Behavior Modeling


Sigmoid function
Scale 1
…Scale 5
Accumulated stimulation by l_q norm pooling
$S_{mr} = \frac{1}{N_{mr}} \left(\sum_{k=1}^{N_{mr}} (s_{mr}(k))^q \right)^{\frac{1}{q}}$
t_0
User i
User i'
t_0
Pattern matching sensor with different temporal resolutions

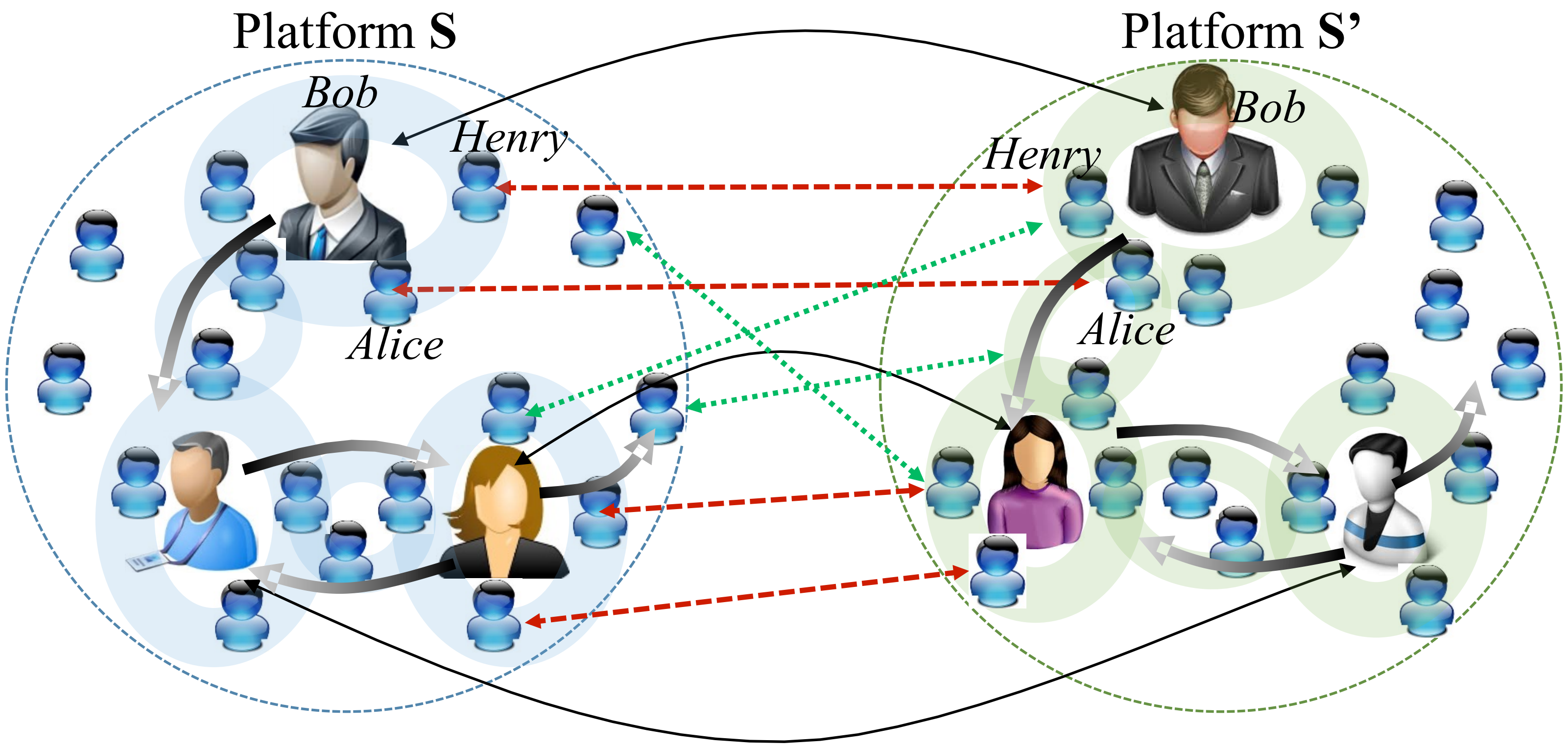
Core Social Network Modeling

- People's closest friends are similar across different social platforms.

- Behavior similarity aggregation of the most frequently interacting friends of users provides insights into user identity linkage.



A Structure Consistency modeling framework



Black arrows: the ground-truth linkage information.
Red arrows: the correct linkage.
Green arrows: the falsely linked persons.

Multi-objective Optimization Framework

- Supervised Learning
- Structure Consistency Modeling
- Multi-objective Optimization

A two-class classification problem --- construct multi-objective optimization which jointly optimizes the **prediction accuracy** on the labeled user pairs and **multiple structure consistency measurements** across different platforms.

Multi-objective Optimization Framework

- Decision Model on Pairwise Similarity

  Support vector machine:

$$f(x) = \mathbf{w}^T x + b$$

$$F_D(\mathbf{w}) = \frac{\gamma_L}{2}||\mathbf{w}||^2 + \sum \xi_{ii'}$$
$$s.t. y_{ii'}(\mathbf{w}^T x_{ii'} + b) \geq 1 - \xi_{ii'}$$

- High Order Structure Consistency

  An eigen-decomposition on the structure consistency graph

$$\max_{\mathbf{w}} \mathbf{w}^T X^T \mathbf{M} X \mathbf{w}$$
$$\Leftrightarrow \min_{\mathbf{w}} \mathbf{w}^T X^T (\mathbf{D} - \mathbf{M}) X \mathbf{w}$$
$$\Leftrightarrow \min_{\mathbf{w}} \mathbf{w}^T X^T \Theta X \mathbf{w}$$
$$s.t. ||\mathbf{w}||^2 \leq s$$

- Multi-objective Optimizatio

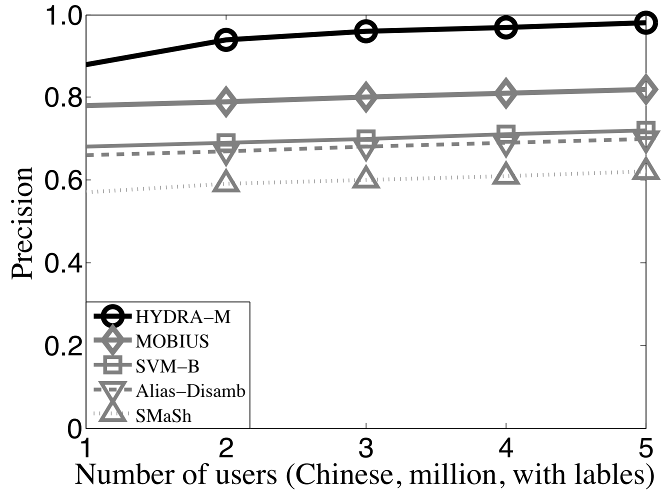
  A generalized semi-supervised learning framework by optimizing the these two objective functions.

$$\min_{\alpha,\beta} \Big\{ \tfrac{1}{2}\alpha^T \left(2\gamma_L \mathbf{K} + \frac{2\gamma_M}{|\mathbb{P}_l \cup \mathbb{P}_u|^2} \mathbf{K}(\mathbf{D} - \mathbf{M})\mathbf{K}\right) \alpha$$
$$- \alpha^T \mathbf{KJY} \beta + \beta^T \mathbf{1} \Big\}$$

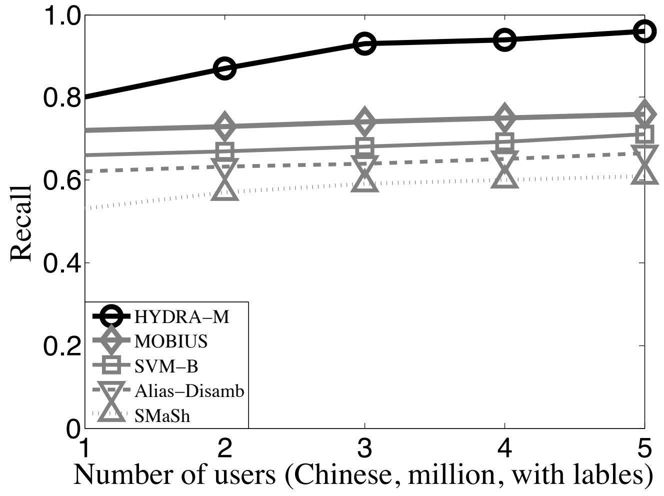
Experiments

1. **Sina Weibo:** (www.weibo.com) A hybrid of Twitter and Facebook with a user base of 500 million users and 47 million daily active users by December 2012.
2. **Tecent Weibo:** (t.qq.com) Another twitter-like micro-blogging service with 500 million users and over 100 million daily active users.
3. **Renren:** (www.renren.com) A social network service dubbed as the Facebook of China with 162 million registered users.
4. **Douban:** (www.douban.com) A social network service for people to share content on topics of movies, books, music, and other off-line events in Chinese cities, with over 100 million monthly unique visitors.
5. **Kaixin:** (www.kaixin001.com) A social network service with 160 million registered users.

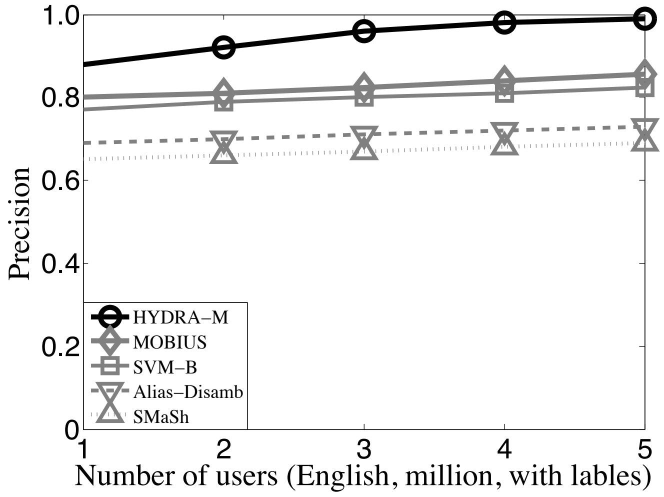
Performance Evaluation



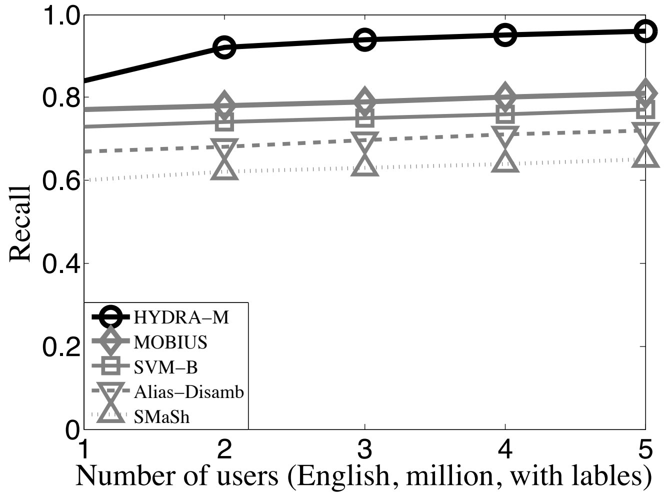
(a) Precision in *Chinese*



(b) Recall in *Chinese*



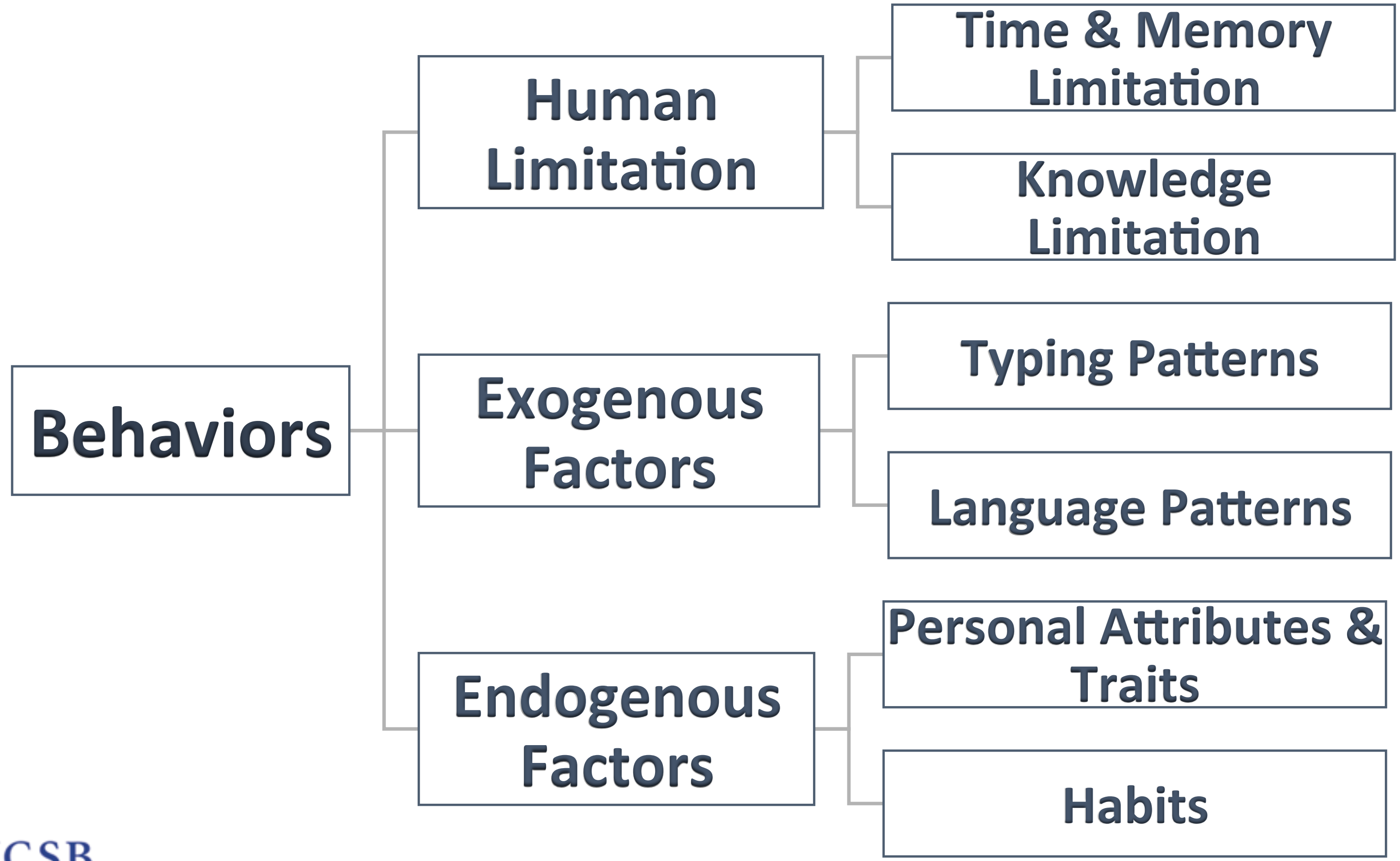
(c) Precision in *English*



(d) Recall in *English*

Connecting Users across Social Media Sites: A Behavioral-Modeling Approach

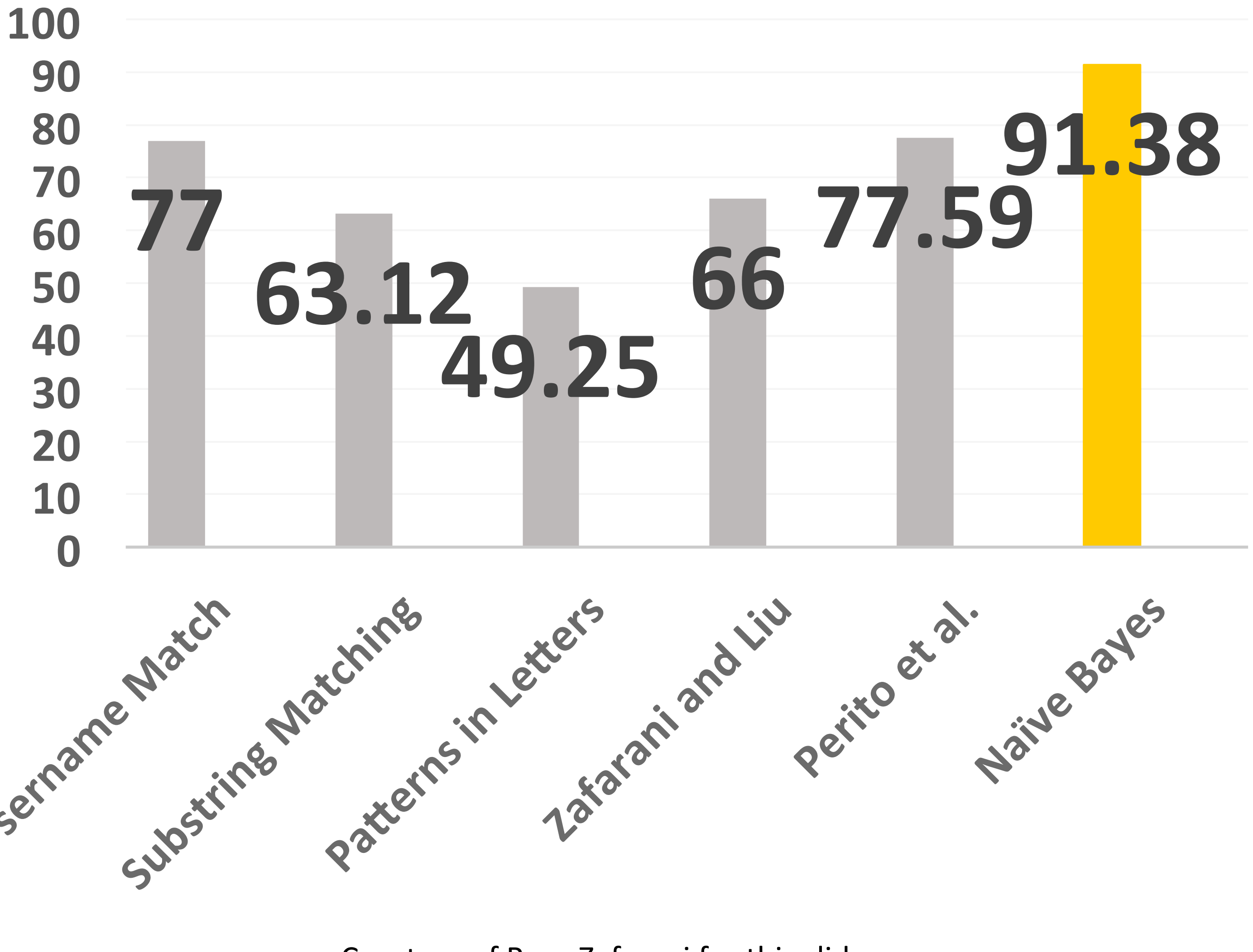
[Zafarani et al. KDD'13]



Courtesy of Reza Zafarani for this slide

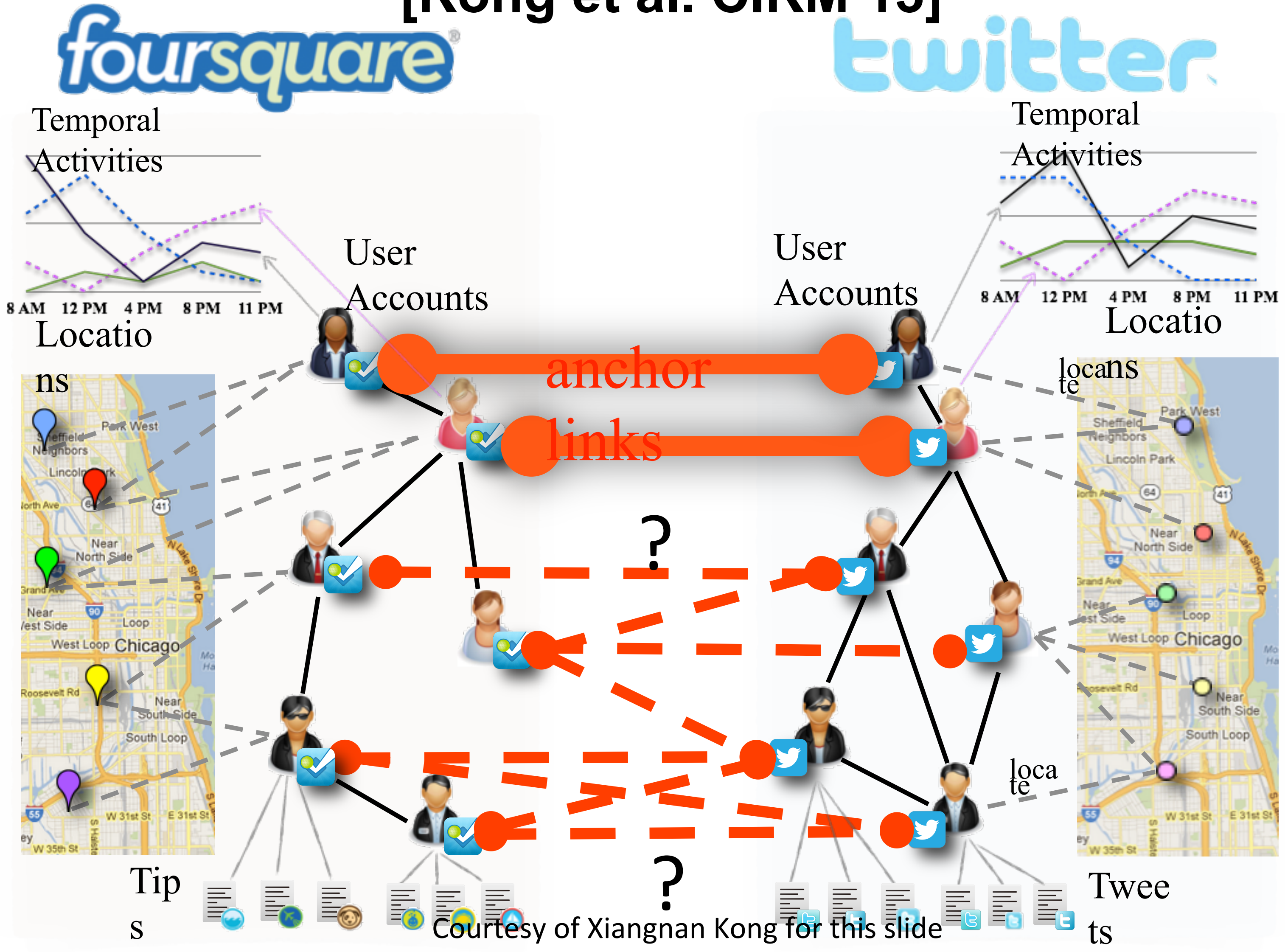
[Zafarani et al. KDD'13]

MOBIUS Performance



Courtesy of Reza Zafarani for this slide

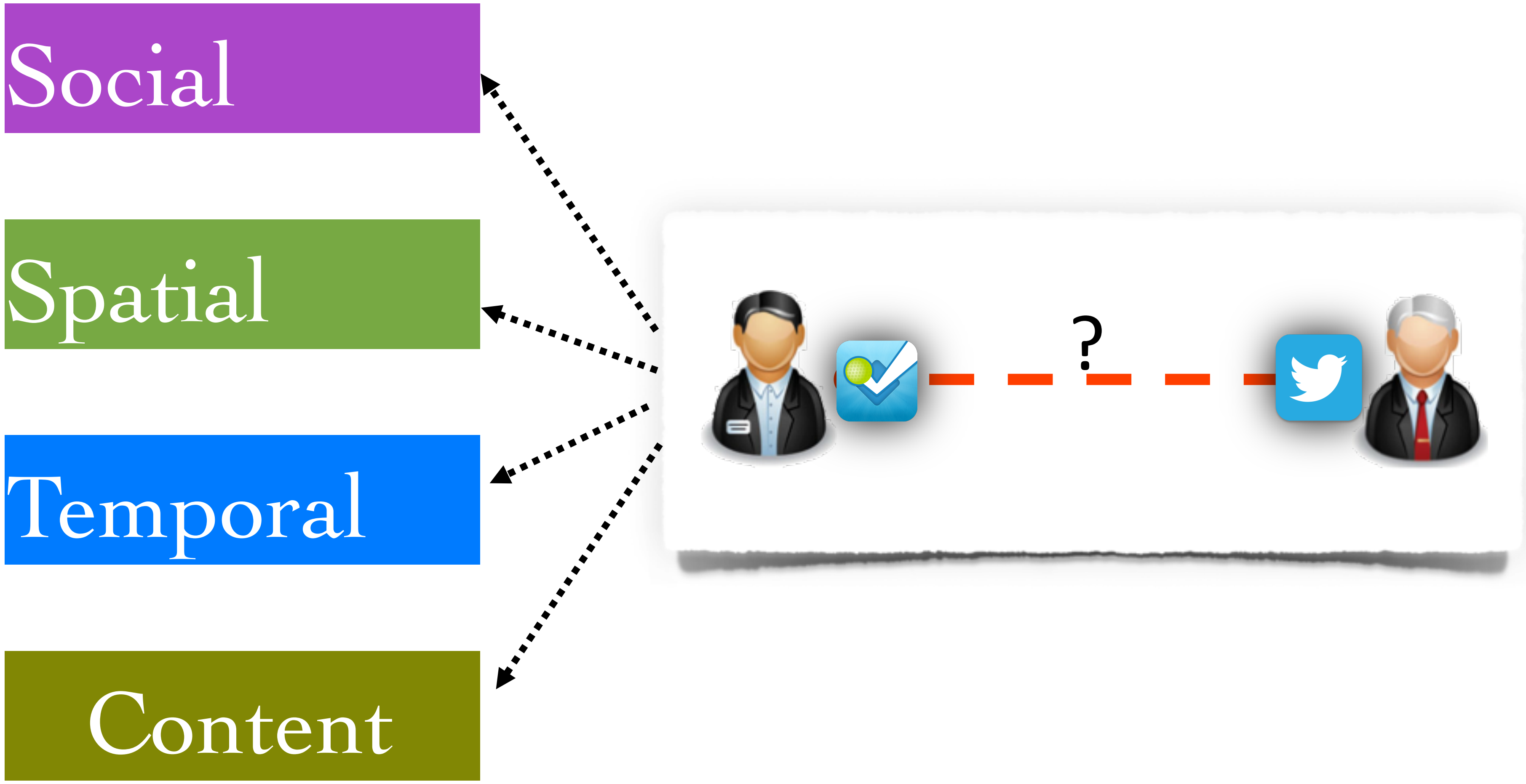
Inferring Anchor Links across Multiple Heterogeneous Social Networks [Kong et al. CIKM'13]



Courtesy of Xiangnan Kong for this slide

[Kong et al. CIKM'13]

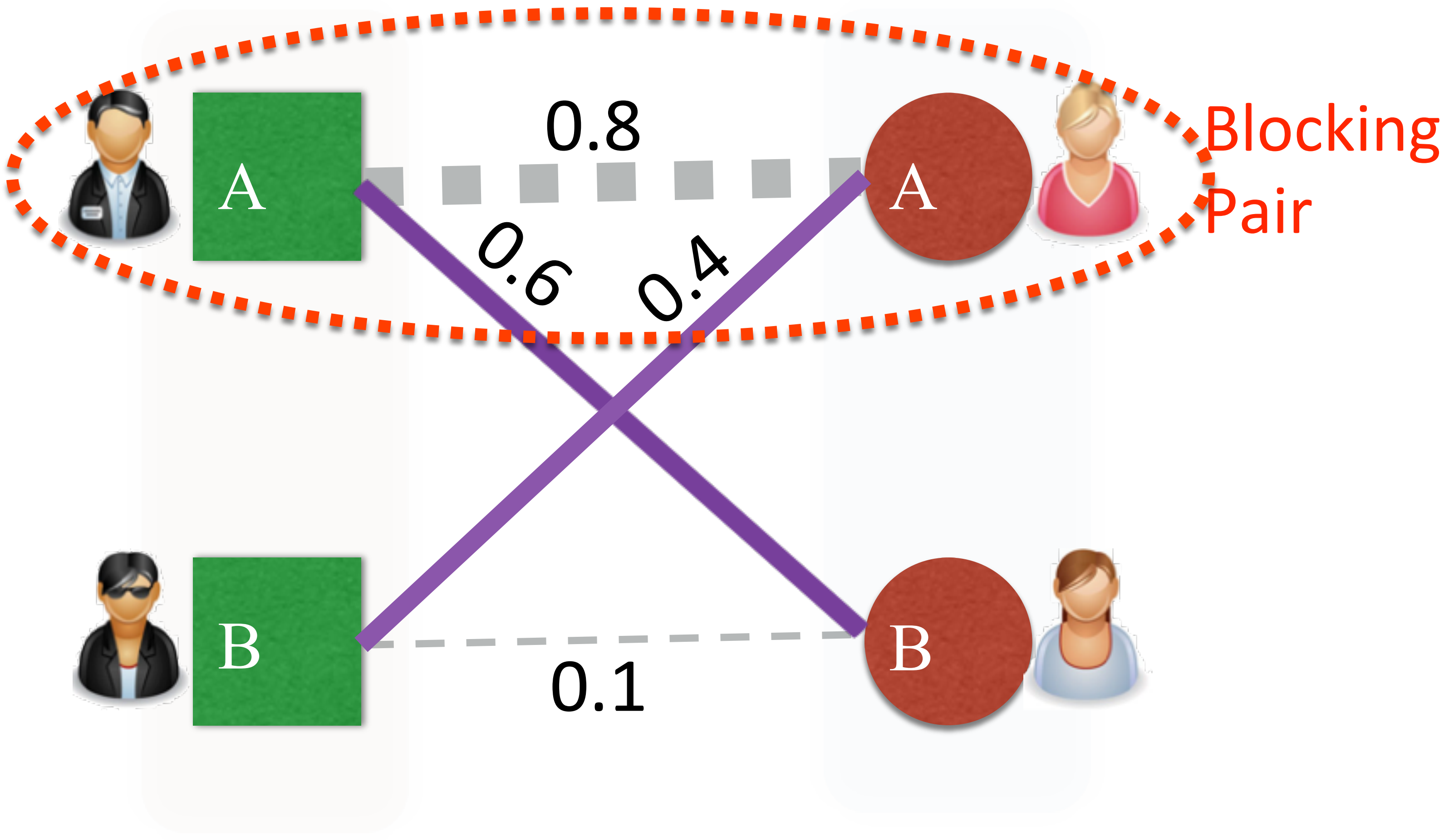
Extract Heterogeneous Cross-Network Features



Courtesy of Xiangnan Kong for this slide

[Kong et al. CIKM'13]

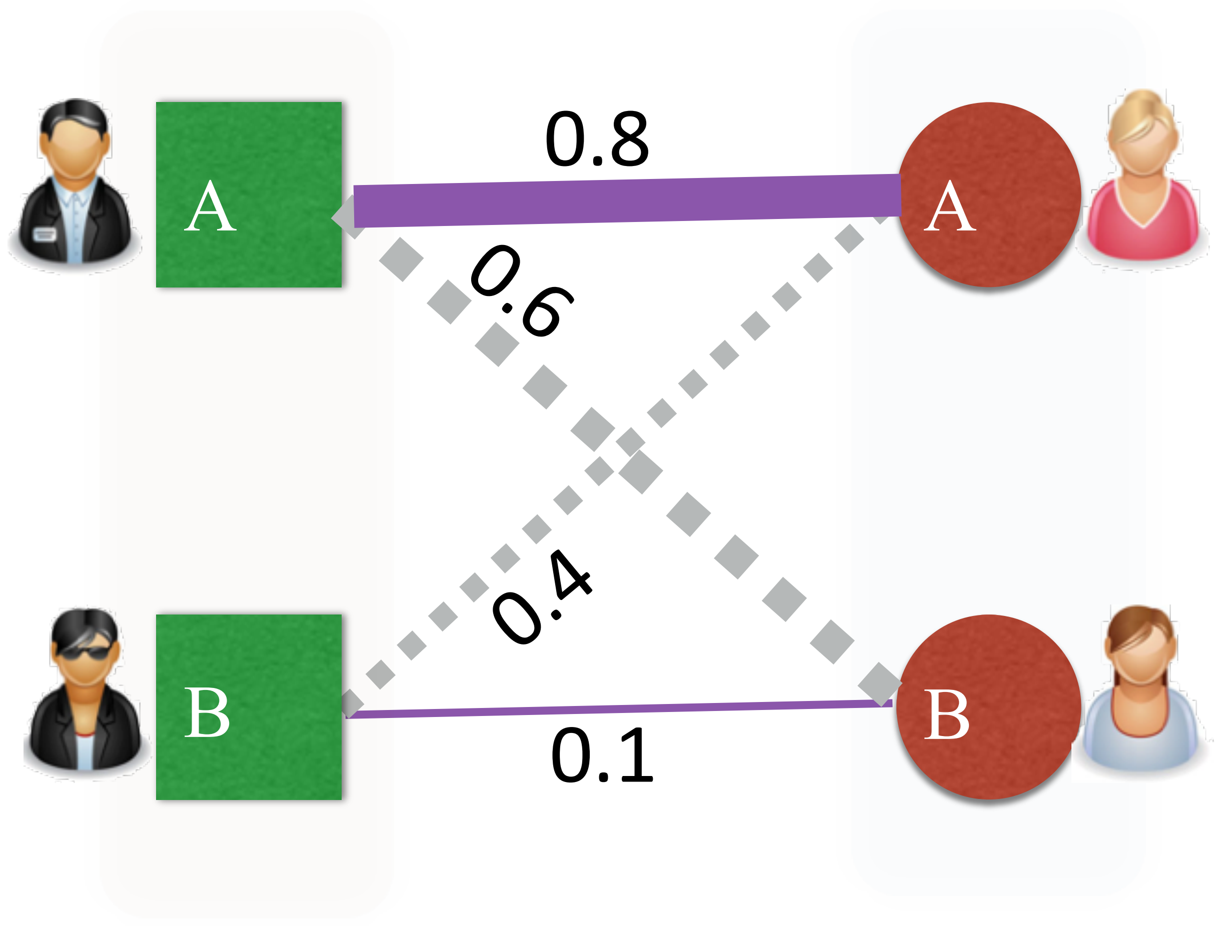
Max Sum of Scores w.r.t. Constraints



Courtesy of Xiangnan Kong for this slide

[Kong et al. CIKM’13]

Stable Matching/Marriage



Courtesy of Xiangnan Kong for this slide

References

- S.Liu, S.Wang, F.Zhu, J.Zhang and R. Krishnan, "***HYDRA: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling***", SIGMOD 2014.
- R. Zafarani and H. Liu, ***"Connecting users across social media sites: A behavioral-modeling approach"***. KDD 2013.
- X. Kong, J. Zhang, and P.S. Yu, "***Inferring Anchor Links across Multiple Heterogeneous Social Networks***", ACM CIKM 2013.
- J. Zhang, X. Kong, and P.S. Yu, "***Predicting Social Links for New Users across Aligned Heterogeneous Social Networks***" IEEE ICDM, 2013.
- J. Zhang, X. Kong, and P.S. Yu, "***Transferring Heterogeneous Links across Location-Based Social Networks***", ACM WSDM 2014.

Conclusion

- Network structure and user behavior play important roles in network mining and analysis for ***social applications***.
- In particular, we have shown how they can be employed to study
    - Network Correlation and Patterns
    - Frequent Network Patterns
    - Collaboration Patterns
- When combined, they can even shed new light into
    - Relationship Mining
    - User Identity Linkage