Social Influence, Community Kernels, and Structural Holes

Jie Tang
Tsinghua University, China

Collaborate with
Jimeng Sun (IBM TJ Watson)
Jiawei Han and Chi Wang (UIUC)
Liaoruo Wang and John Hopcroft (Cornell)
Zi Yang (CMU), Lu Liu and Tiancheng Lou (THU)
Social Networks

• >900 million users
• the 3rd largest “Country” in the world
• More visitors than Google

• >500 million users

• More than 5 billion images

• 2009, 2 billion tweets per quarter
• 2010, 4 billion tweets per quarter
• 2011, 25 billion tweets per quarter

• 2012, 300 million users, 300% yearly increase

• Pinterest, with a traffic higher than Twitter and Google
Social Networks

- >900 million users
- the 3rd largest “Country” in the world
- More visitors than Google

Social networks already become a bridge to connect our daily physical life and the virtual web space

O2O!

- 2012, 300 million users, 300% yearly increase
- Pinterest, with a traffic higher than Twitter and Google
What is a social network?

- A **social network** is:
  - a **graph** made up of:
  - a set of **individuals**, called “nodes”, and
  - tied by one or more **interdependency**, such as friendship, called “edges”.
What are the **fundamentally new** things in social networks?

Web 1.0

(1) links between data points
(2) the network can be heterogeneous
What are the fundamentally new things in social networks?

Collaborative Web

(1) personalized learning
(2) collaborative filtering
What are the **fundamentally new** things in social networks?

Social Web

(1) interactions
(2) information diffusion

Collective intelligence
Find $K$ nodes (users) in a social network that could maximize the spread of influence (Domingos, 01; Richardson, 02; Kempe, 03)
Example—opinion leaders

Who are the opinion leaders in a community?

Questions:
- How to quantify the strength of social influence between users?
- How to predict users’ behaviors over time?

spread of influence (Domingos, 01; Richardson, 02; Kempe, 03)
Example—structural holes

Structural hole users control the information flow between different communities (Burt, 92; Podolny, 97; Ahuja, 00; Kleinberg, 08)
Motivation

• **Modeling** the influence between users and the network structural formation becomes a very important issue and can benefit many real applications
  – Advertising
  – Social recommendation
  – Marketing
  – Social security
  – …
Social Influence Analysis

- Lu Liu, Jie Tang, Jiawei Han, and Shiqiang Yang. Learning Influence from Heterogeneous Social Networks. *Data Mining and Knowledge Discovery*. (to appear)
- Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining Topic-Level Influence in Heterogeneous Networks. *CIKM 2010*. pp. 199-208.
Learning topic-based influence between users

- Social network -> Topical influence network

### Input: coauthor network

- Topics:
  - Topic 1: Data mining
  - Topic 2: Database

### Social influence analysis

- Node factor function: $g(v_1, y_1, z)$
- Edge factor function: $f(y_i, y_j, z)$

### Output: topic-based social influences

- Topics:
  - Topic 1: Data mining
  - Topic 2: Database
Several key challenges:

• How to differentiate the social influences from different angles (topics)?
• How to incorporate different information (e.g., topic distribution and network structure) into a unified model?
• How to estimate the model on real-large networks?
Our Solution: Topical Affinity Propagation

- Topical Affinity Propagation
  - Topical Factor Graph model
  - Efficient learning algorithm
  - Distributed implementation
Topical Factor Graph (TFG) Model

The problem is cast as identifying which node has the highest probability to influence another node on a specific topic along with the edge.
The learning task is to find a configuration for all \( \{y_i\} \) to maximize the joint probability.

Objective function:

\[
P(v, Y) = \frac{1}{Z} \prod_{i=1}^{N} \prod_{z=1}^{T} \prod_{k=1}^{T} h(y_1, \ldots, y_N, k, z) \prod_{i=1}^{N} \prod_{z=1}^{T} g(v_i, y_i, z) \prod_{e_{kl} \in E} \prod_{z=1}^{T} f(y_k, y_l, z)
\]

1. How to define?
2. How to optimize?
How to define (topical) feature functions?

– Node feature function

\[
g(v_i, y_i, z) = \begin{cases} 
\frac{w_{i,y_i}^z}{\sum_{j \in NB(i)} (w_{i,j}^z + w_{j,i}^z)} & y_i^z \neq i \\
\frac{w_{j,i}^z}{\sum_{j \in NB(i)} (w_{i,j}^z + w_{j,i}^z)} & y_i^z = i 
\end{cases}
\]

– Edge feature function

\[
f(y_i, y_j) = \begin{cases} 
w[y_i \sim y_j] & y_i = y_j \\
1 - w[y_i \sim y_j] & y_i \neq y_j 
\end{cases}
\]

or simply binary

– Global feature function

\[
h(y_1, \ldots, y_N, k, z) = \begin{cases} 
0 & \text{if } y_k^z = k \text{ and } y_i^z \neq k \text{ for all } i \neq k \\
1 & \text{otherwise.}
\end{cases}
\]
Model Learning Algorithm

\[ m_{y_i \rightarrow f_{ij}(y_i)} = \prod_{f' \sim y_i \setminus f_{ij}} m_{f' \rightarrow y_i(y_i)} \]

Sum-product:

\[ m_{f_{ij} \rightarrow y_i(y_i)} = \sum_{y_i \sim \{y_i\}} \left( \prod_{y' \sim f \setminus y_i} f(y_i, y') m_{y' \rightarrow f_{ij}(y')} \right) \]

- Low efficiency!
- Not easy for distributed learning!
New TAP Learning Algorithm

1. Introduce two new variables $r$ and $a$, to replace the original message $m$.

2. Design new update rules:

\[
\begin{align*}
    r_{ij} &= b_{ij} - \max_{k \in NB(j)} \left\{ b_{ik} + a_{ik} \right\} \\
    a_{jj} &= \max_{k \in NB(j)} \min \left\{ r_{kj}, 0 \right\} \\
    a_{ij} &= \min(\max \left\{ r_{jj}, 0 \right\}, -\min \left\{ r_{jj}, 0 \right\} \\
    &- \max_{k \in NB(j) \setminus \{i\}} \min \left\{ r_{kj}, 0 \right\}), \quad i \in NB(j)
\end{align*}
\]
The TAP Learning Algorithm

Input: $G = (V, E)$ and topic distributions $\{\theta_v\}_{v \in V}$
Output: topic-level social influence graphs $\{G_z = (V_z, E_z)\}_T$

1.1 Calculate the node feature function $g(v_i, y_i, z)$;
1.2 Calculate $b_{ij}^z$ according to Eq. 8;
1.3 Initialize all $\{r_{ij}^z\} \leftarrow 0$;
1.4 repeat
   1.5 foreach edge-topic pair $(e_{ij}, z)$ do
      1.6 Update $r_{ij}^z$ according to Eq. 5;
   1.7 end
   1.8 foreach node-topic pair $(v_j, z)$ do
      1.9 Update $a_{jj}^z$ according to Eq. 6;
   1.10 end
   1.11 foreach edge-topic pair $(e_{ij}, z)$ do
      1.12 Update $a_{ij}^z$ according to Eq. 7;
   1.13 end
   1.14 until convergence;
1.15 foreach node $v_t$ do
   1.16 foreach neighboring node $s \in NB(t) \cup \{t\}$ do
      1.17 Compute $\mu_{st}^z$ according to Eq. 9;
   1.18 end
1.19 end
1.20 Generate $G_z = (V_z, E_z)$ for every topic $z$ according to $\{\mu_{st}^z\}$;

$b_{ij}^z = \log \frac{g(v_i, y_i, z) | y_i^z = j}{\sum_{k \in NB(i) \cup \{i\}} g(v_i, y_i, z) | y_i^z = k}$

$r_{ij}^z = b_{ij}^z - \max_{k \in NB(j)} \{b_{ik}^z + a_{ik}^z\}$

$a_{jj}^z = \max_{k \in NB(j)} \min \{r_{kj}^z, 0\}$

$a_{ij}^z = \min(\max \{r_{jj}^z, 0\}, -\min \{r_{jj}^z, 0\} - \max_{k \in NB(j) \setminus \{i\}} \min \{r_{kj}^z, 0\}), i \in NB(j)$

$\mu_{st}^z = \frac{1}{1 + e^{-(r_{ts}^z + a_{ts}^z)}}$
Distributed TAP Learning

- **Map-Reduce**
  - **Map**: (key, value) pairs
    - $e_{ij}/a_{ij} \rightarrow e_{i*}/a_{ij}$; $e_{ij}/b_{ij} \rightarrow e_{i*}/b_{ij}$; $e_{ij}/r_{ij} \rightarrow e_{*j}/r_{ij}$.
  - **Reduce**: (key, value) pairs
    - $e_{ij}/* \rightarrow$ new $r_{ij}$; $e_{ij}/* \rightarrow$ new $a_{ij}$

- **For the global feature function**

  **Theorem 1.** If the global feature function $h$ can be factorized into $h = \prod_{k=1}^{N} h_k$, for every $i \in \{1, \ldots, N\}$, $y_i \neq k$, $y'_i \neq k$, $h_k(y_1, \ldots, y_i, \ldots, y_N) = h_k(y_1, \ldots, y'_i, \ldots, y_N)$, then the message passing update rules can be simplified to influence update rules. \[\blacksquare\]
Experiment

• Data set: (http://arnetminer.org/lab-datasets/soinf/)

<table>
<thead>
<tr>
<th>Data set</th>
<th>#Nodes</th>
<th>#Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coauthor</td>
<td>640,134</td>
<td>1,554,643</td>
</tr>
<tr>
<td>Citation</td>
<td>2,329,760</td>
<td>12,710,347</td>
</tr>
<tr>
<td>Film</td>
<td>18,518 films 7,211 directors 10,128 actors 9,784 writers</td>
<td>142,426</td>
</tr>
</tbody>
</table>

• Evaluation measures
  – CPU time
  – Case study
  – Application
Table 2: Scalability performance of different methods on real data sets. >10hr means that the algorithm did not terminate when the algorithm runs more than 10 hours.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Citation</th>
<th>Coauthor</th>
<th>Film</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Product</td>
<td>N/A</td>
<td>&gt;10hr</td>
<td>1.8 hr</td>
</tr>
<tr>
<td>Basic TAP Learning</td>
<td>&gt;10hr</td>
<td>369s</td>
<td>57s</td>
</tr>
<tr>
<td>Distributed TAP Learning</td>
<td>39.33m</td>
<td>104s</td>
<td>148s</td>
</tr>
</tbody>
</table>
Speedup results

Speedup vs. Dataset size

Perfect
Our method

Speedup vs. #Computer nodes
## Influential nodes on different topics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Topic</th>
<th>Representative Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Mining</td>
<td>Heikki Mannila, Philip S. Yu, Dimitrios Gunopulos, Jiawei Han, Christos Faloutsos, Bing Liu, Vipin Kumar, Tom M. Mitchell, Wei Wang, Qiang Yang, Xindong Wu, Jeffrey Xu Yu, Osmar R. Zaiane</td>
<td></td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Pat Langley, Alex Waibel, Trevor Darrell, C. Lee Giles, Terrence J. Sejnowski, Samy Bengio, Daphne Koller, Luc De Raedt, Vasant Honavar, Floriana Esposito, Bernhard Scholkopf</td>
<td></td>
</tr>
<tr>
<td>Database System</td>
<td>Gerhard Weikum, John Mylopoulos, Michael Stonebraker, Barbara Pernici, Philip S. Yu, Sharad Mehrotra, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han</td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td>Yan Wang, Liang-je Zhang, Schahram Dustdar, Jian Yang, Fabio Casati, Wei Xu, Zakaria Maamar, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah</td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>Wolfgang Nejdl, Daniel Schwabe, Steffen Staab, Mark A. Musen, Andrew Tomkins, Juliana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta</td>
<td></td>
</tr>
<tr>
<td>Bayesian Network</td>
<td>Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Didier Dubois, David Heckerman, Philippe Smets</td>
<td></td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Mining</td>
<td>Fast Algorithms for Mining Association Rules in Large Databases, Using Segmented Right-Deep Trees for the Execution of Pipelined Hash Joins, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Discovery of Multiple-Level Association Rules from Large Databases, Interleaving a Join Sequence with Semijoins in Distributed Query Processing</td>
<td></td>
</tr>
<tr>
<td>Semantic Web</td>
<td>FaCT and iFaCT, The GRAIL concept modeling language for medical terminology, Semantic Integration of Semistructured and Structured Data Sources, Description of the RACER System and its Applications, DL-Lite: Practical Reasoning for Rich Dis</td>
<td></td>
</tr>
</tbody>
</table>
Social Influence Sub-graph on “Data mining”
Application—Expert Finding

Table 7: Performance of expert finding with different approaches.

Expert finding data from (Tang, KDD08; ICDM08)
http://arnetminer.org/lab-datasets/expertfinding/
Community Kernels

Pareto Principle: Less than 1% of the Twitter users (e.g. entertainers, politicians, writers) produce 50% of its content, while the others (e.g. fans, followers, readers) have much less influence and completely different social behavior.

Kernel users and ordinary users exhibit very different behaviors.
Approach (Greedy & WeBA)

- **Challenges**
  - How to identify kernel members, and
  - How to determine the structural of community kernels.

- **Formalize the problem into an optimization problem.**
  - Proposed two new algorithms
    - Greedy
      - Maximum cardinality search (MCS)
      - Runs in linear time
    - Global Relaxation
      - Random walk (Annealing)
      - Theoretical error bound
      - Almost linear time.
Unbalanced Weakly-Bipartite Structure

Empirical property of many real-world networks:

\[ d_{21} > d_{11} > d_{22} \gg d_{12} \]

\[ d_{ij} = \frac{|E(V_i, V_j)|}{|V_j|}, \quad i, j \in \{1, 2\} \]

<table>
<thead>
<tr>
<th>Network</th>
<th>(d_{21})</th>
<th>(d_{11})</th>
<th>(d_{22})</th>
<th>(d_{12})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coauthor</td>
<td>14.19</td>
<td>5.34</td>
<td>4.42</td>
<td>0.37</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1689.31</td>
<td>104.22</td>
<td>4.69</td>
<td>0.60</td>
</tr>
<tr>
<td>Twitter</td>
<td>110.78</td>
<td>26.78</td>
<td>2.94</td>
<td>0.29</td>
</tr>
<tr>
<td>Slashdot</td>
<td>180.90</td>
<td>84.56</td>
<td>10.75</td>
<td>0.64</td>
</tr>
<tr>
<td>Citation</td>
<td>76.69</td>
<td>35.81</td>
<td>23.80</td>
<td>0.26</td>
</tr>
</tbody>
</table>
GREEDY ALGORITHM

- Given an graph $G = (V, E)$ and a kernel size $k$
  - Initialize the set $S$ to be a random vertex $v \in V$
  - Iteratively add to $S$ the vertex with the most connections to $S$
  - Always pick the vertex with the highest degree

- Example
**Greedy Algorithm**

- Given an graph $G = (V, E)$ and a kernel size $k$
  - Initialize the set $S$ to be a random vertex $v \in V$
  - Iteratively add to $S$ the vertex with the most connections to $S$
  - Always pick the vertex with the highest degree

- Running time and space complexity: $O(|V| + |E|)$
- No guaranteed error bound
- Repeat $O(|V|/k)$ times to obtain steady state and reduce the effect of random selection of the initial point
WEIGHT-BALANCED ALGORITHM (WeBA)

- Each vertex $v \in V$ has a weight vector $\vec{w}(v) = \{w_1(v), \ldots, w_l(v)\}$ to represent its relative importance for each community kernel.

- Optimization Problem:

$$\max \quad \mathcal{L}(\vec{w}) = \sum_{(u,v) \in E} \vec{w}(u) \cdot \vec{w}(v)$$

subject to

$$\sum_{v \in V} w_i(v) = k, \quad \forall i \in \{1, \ldots, l\}$$

$$\sum_{1 \leq i \leq l} w_i(v) \leq 1, \quad \forall v \in V$$

$$w_i(v) \geq 0, \quad \forall v \in V, \quad \forall i \in \{1, \ldots, l\}$$

- Intractable to solve — we approximate the solution by iteratively solving its one-dimensional version $\mathcal{L}(w)$.
WEIGHT-BALANCED ALGORITHM (WEBA)

- **Theorem 1**: A global maximum of the objective function $\mathcal{L}(w)$ corresponds to a community kernel.

- Given an graph $G = (V, E)$ and a kernel size $k$, maximizing $\mathcal{L}(w)$ is NP-hard.
  - Initialize the set $S$ to be a random subset obtained by GREEDY
  - Assign weight 1 to each vertex in $S$ and weight 0 otherwise
  - If $\exists u, v \in V$ such that $w(u) < 1, w(v) > 0$ and $nw(u) > nw(v)$, where $nw(u)$ is the neighboring weight of $u$, the weights of $u$ and $v$ are modified to locally maximize $\mathcal{L}(w)$

relaxation conditions
Input: $G = (V, E)$ and kernel size $k$
Output: community kernels $K = \{K_1, K_2, \cdots, K_\ell\}$

$K \leftarrow \emptyset$
repeat
    $S \leftarrow$ a subset returned by GREEDY($G, k$)
    $\forall v \in S$, $w(v) \leftarrow 1$; $\forall v \notin S$, $w(v) \leftarrow 0$
    while $\exists u, v \in V$ satisfying the relaxation conditions do
        if $(u, v) \notin E$ then $\delta \leftarrow \min \{1 - w(u), w(v)\}$
        else $\delta \leftarrow \min \left\{ 1 - w(u), w(v), \frac{nw(u) - nw(v)}{2} \right\}$
        pick one pair $\{u, v\}$ with the maximum $\delta$ value
        $w(u) \leftarrow w(u) + \delta$, $w(v) \leftarrow w(v) - \delta$
    end
    $C \leftarrow \{v \in V \mid w(v) = 1\}$
    if $C \notin K$ then $K \leftarrow \{K, C\}$
until $O(|V|/k)$ times;
return $K$
An Example

- Given a graph and a kernel size $k = 3$
- Given a random subset of size $k$
WEIGHT-BALANCED ALGORITHM (WEBA)

- Three pairs of vertices satisfy the relaxation conditions with the maximum $\delta = 1$
**WEIGHT-BALANCED ALGORITHM (WeBA)**

- \( w(u) \leftarrow w(u) + \delta \quad \rightarrow \quad w(u) \leftarrow 1 \)
- \( w(v) \leftarrow w(v) - \delta \quad \rightarrow \quad w(v) \leftarrow 0 \)

![Graph of the weight-balanced algorithm](image)
WEIGHT-BALANCED ALGORITHM (WeBA)

• Keep balancing weights as described above until no pairs of vertices satisfy the relaxation conditions
WEBA

- **Theorem 2 (correctness):**
  WEBA is guaranteed to converge to a feasible solution.

- **Theorem 3 (error bound):**
  For any assigned weights \(\{w(v), \forall v \in V\}\) and any \(\varepsilon > 0\), after
  \[
  \max \left\{ \frac{4k^3D^5}{\varepsilon^2}, \frac{2mkD^3}{\varepsilon} \right\}
  \]
  iterations, we have \(\mathcal{L}(w^*(v)) - \mathcal{L}(w(v)) \leq \varepsilon\).

- Repeat \(O(|V|/k)\) times to obtain steady state and reduce the effect of random selection of the initial point.
FINDING AUXILIARY COMMUNITY

- Given community kernels $\{K_1, K_2, \ldots, K_l\}$
  - Label each vertex that is not in any kernel as unassociated
  - For each unassociated vertex, rank the kernels according to the number of edges from the vertex to each kernel and the vertices that have already been associated with that kernel
    - Associate the vertex with the top-ranked kernel(s)
    - Repeat this process until no more vertices can be associated
- Auxiliary communities can overlap with each other
FINDING AUXILIARY COMMUNITY
EXPERIMENTAL RESULTS

• Data Sets
  – Coauthor (822,415 nodes; 2,928,360 edges)
    • Benchmark coauthor network (52,146 nodes; 134,539 edges)
  – Wikipedia (310,990 nodes; 10,780,996 edges)
    • Namespace talk pages (263 nodes; 1,075 edges)
    • User personal pages (266 nodes; 33,829 edges)
  – Twitter (465,023 nodes; 833,590 edges)

• Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Local Spectral Partitioning (LSP)</th>
<th>METIS+MQI</th>
<th>d-LSP (high-degree)</th>
<th>NEWMAN1 (betweenness)</th>
<th>p-LSP (high-PageRank)</th>
<th>NEWMAN2 (modularity)</th>
<th>α-β</th>
<th>LOUVAIN</th>
</tr>
</thead>
</table>
CASE STUDY ON TWITTER

Community Kernels by WEBA

Community Structure by NEWMAN2

ENTERTAINERS
Ashton Kutcher  Demi Moore
Oprah Winfrey  Jimmy Fallon
Rob Riggle  Al Yankovic

POLITICIANS
Sarah Palin  Karl Rove
John McCain  Mike Huckabee
Mitt Romney  Tim Huelskamp

Community Structure by METIS+MQI
Results on Coauthor & Wikipedia

- On average, WeBA improves Precision by 340% (wiki) and 70% (coauthor), and improves Recall by 130% (wiki) and 41% (coauthor).

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wiki</td>
<td>coauthor</td>
</tr>
<tr>
<td></td>
<td>Talk</td>
<td>User</td>
</tr>
<tr>
<td>LSP</td>
<td>0.061</td>
<td>0.085</td>
</tr>
<tr>
<td>d-LSP</td>
<td>0.051</td>
<td>0.091</td>
</tr>
<tr>
<td>p-LSP</td>
<td>0.046</td>
<td>0.082</td>
</tr>
<tr>
<td>Metis+MQI</td>
<td>0.049</td>
<td>0.012</td>
</tr>
<tr>
<td>Louvain</td>
<td>0.063</td>
<td>0.122</td>
</tr>
<tr>
<td>Newman1</td>
<td>0.033</td>
<td>0.203</td>
</tr>
<tr>
<td>Newman2</td>
<td>0.039</td>
<td>0.085</td>
</tr>
<tr>
<td>α-β</td>
<td>0.324</td>
<td>0.336</td>
</tr>
</tbody>
</table>

- **WeBA**: 0.456 0.46 **0.852** ... **0.837** **0.911**
- **Greedy**: 0.334 0.403 0.83 ... 0.746 0.752

The highlighted values indicate improvements over traditional methods.
On average, WEBA increases F1-score by **300%** (wiki) and **61%** (coauthor), and increases Resemblance by **180%** (wiki) and **67%** (coauthor).

<table>
<thead>
<tr>
<th></th>
<th>F1-score</th>
<th>Resemblance (Jaccard Index)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>wik</td>
</tr>
<tr>
<td></td>
<td>Talk</td>
<td>User</td>
</tr>
<tr>
<td>LSP</td>
<td>0.090</td>
<td>0.134</td>
</tr>
<tr>
<td>d-LSP</td>
<td>0.091</td>
<td>0.137</td>
</tr>
<tr>
<td>p-LSP</td>
<td>0.083</td>
<td>0.121</td>
</tr>
<tr>
<td>METIS+MQI</td>
<td>0.055</td>
<td>0.023</td>
</tr>
<tr>
<td>LOUVAIN</td>
<td>0.108</td>
<td>0.181</td>
</tr>
<tr>
<td>NEWMAN1</td>
<td>0.014</td>
<td>0.111</td>
</tr>
<tr>
<td>NEWMAN2</td>
<td>0.033</td>
<td>0.080</td>
</tr>
<tr>
<td>α-β</td>
<td>0.367</td>
<td>0.376</td>
</tr>
<tr>
<td><strong>WEBA</strong></td>
<td><strong>0.514</strong></td>
<td><strong>0.509</strong></td>
</tr>
<tr>
<td><strong>GREEDY</strong></td>
<td>0.377</td>
<td>0.446</td>
</tr>
</tbody>
</table>
SENSITIVITY

(a) Precision vs. Recall

(b) F1-score vs. kernel size
E  F  I  C  I  E  N  C  Y  —  T  W  I  T  T  E  R  &  C  O  A  U  T  H  O  R

465,023 nodes, 833,590 edges

822,415 nodes, 2,928,360 edges
Top-\(k\) Structural Holes

-Tiancheng Lou and Jie Tang. Mining Top-\(k\) Structural Holes in Large Networks. (Submitted)
The **structural hole** theory suggests that:

- Individuals would benefit from filling the “holes” between **communities** that are otherwise disconnected.
- Structural holes play a key role in the **information diffusion**.

Lack of a principled methodology to discover structural holes from a given social network.
Problem Definition

• Top-

-k Structural Holes Detection.
  – Denote $G=(V,E)$ as a social network, with $l$ communities $\mathcal{C}=(C_1,C_2,\ldots,C_l)$
  – Denote top-

-k structural holes $V_{SH}$ as a subset of $k$ nodes, which maximize the following qualify function:
    \[ \max Q(V_{SH}, \mathcal{C}), \text{ with } |V_{SH}|=k \]
  – Since the utility function $Q$ is a general definition, which can be instantiated in different ways,

• Our contribution
  – We develop two instantiation models based on the above objective function.
Model 1 : HIS

• Opinion leader vs. Structural holes

\[ I(v, i) = \max_{(u,v) \in E, i \in S} \{ I(v, i), \alpha_i I(u, i) + \beta S H(u, S) \} \]

\[ H(v, S) = \min_{i \in S} \{ I(v, i) \} \]

• Analyze
  – Condition of existence
  – \(\varepsilon\)- convergence property
  – Iterative algorithm and improved algorithm
  – Parameter analysis
Model 2 : MaxD

• Define $Q(V_{SH}, C) = MC(G, C) - MC(G \setminus V_{SH}, C)$
  – Where $MC(G, C)$ is the minimal cut of communities $C$ in $G$.

• Analyze
  – NP-Hardness of the exact solution, the first attempt to proof minimal node-cut problem in an un-weighted graph.
  – Suppose the k-DENSEST SUBGRAPH is hard to approximate within $n^{\Omega(1)}$, then the problem is also hard to approximate within $n^{\Omega(1)}$ as well.
  – Approximation algorithm and evaluation methods.
Data Sets

• **Coauthors:** 815,946 authors and 2,792,833 coauthorships
  – Six areas: Artificial Intelligence (AI), Databases (DB), Data Mining (DM), Distributed Parallel Computing (DP), Graphics, Vision and HCI (GV), as well as Networks, Communications and Performance (NC)
  – View PC members in more than one areas as spanning structural holes

• **Twitter:** 13,442,659 users and 56,893,234 links
  – Communities are discovered by the community kernel detection algorithm

• **Inventor:** 2,445,351 inventors and 5,841,940 co-inventing relationships
  – Each company is a community
Observation 1

Structural hole nodes are more likely to connect importance nodes than opinion leaders.
Observation 2

Structural hole nodes receive more cross-domain citations than opinion leaders.

(a) Cross domain

(b) Outer domain
Observation 3

Opinion leaders play a key role in spreading information within a community, while structural hole nodes are more important for spreading information between communities.
Results on Coauthor

- Use Coauthor as the benchmark dataset to evaluate the accuracy of the proposed models.
  - Both models clearly outperform the comparison algorithms by +20 – 40%.
  - Structural holes are determined by not only bridging positions, but also status of neighbors.
Results on Twitter

• Information spread between different communities.
  – On Twitter: top 0.2% structural hole users almost influence 10% of the tweets-forwarding behaviors between different communities.
  – On Coauthor: Improvement is statistically significant ($p << 0.01$)
Case Study on Inventor Network

- Most of the detected structural holes have been working in more than one job.

Table 2: Top-5 structural hole nodes discovered by our algorithms on the Inventor network. Names with * are inventors with the highest (top-5) PageRank scores.

<table>
<thead>
<tr>
<th>Inventor</th>
<th>HIS</th>
<th>MaxD</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. Boyden</td>
<td>✓</td>
<td></td>
<td>Professor (MIT Media Lab)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓</td>
<td>Associate Professor (MIT McGovern Inst.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Group Leader (Synthetic Neurobiology)</td>
</tr>
<tr>
<td>A. Czarnik</td>
<td>✓</td>
<td></td>
<td>Founder and Manager (Protia, LLC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Visiting Professor (University of Nevada)</td>
</tr>
<tr>
<td>T. Kondo*</td>
<td>✓</td>
<td>✓</td>
<td>Co-Founder (Chief Scientific Officer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Senior vice president (Sony Corporation)</td>
</tr>
<tr>
<td>A. Nishio</td>
<td>✓</td>
<td></td>
<td>Director of Operations (WBI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Director of Department Responsible (IDA)</td>
</tr>
<tr>
<td>E. Nowak*</td>
<td>✓</td>
<td></td>
<td>Senior vice President (Walt Disney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Secretary of Trustees (The New York Eye)</td>
</tr>
<tr>
<td>A. Rofougaran</td>
<td>✓</td>
<td></td>
<td>Consultant (various wireless companies)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Co-founder (Innovent System Corp.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Leader (RF-CMOS).</td>
</tr>
<tr>
<td>M. Rofougaran</td>
<td>✓</td>
<td>✓</td>
<td>Engineering Director (Broadcom Corp.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Co-founder(Iran Today Publications)</td>
</tr>
<tr>
<td>S. Yamazaki*</td>
<td>✓</td>
<td></td>
<td>President and majority shareholder (SEL)</td>
</tr>
</tbody>
</table>
Thank you!

QA?

Data & Code:
http://arnetminer.org/lab-datasets/soinf
http://arnetminer.org/stnt