INCREASING THE SIZE OF PARTIALLY OBSERVED NETWORKS

Sucheta Soundarajan  
Syracuse Univ.

Tina Eliassi-Rad  
Northeastern Univ.

Brian Gallagher  
LLNL

Ali Pinar  
Sandia Labs
Incomplete Networks

- Networked representations of physical and social phenomena are often incomplete because the phenomena is partially observed.
- Working with incomplete networks can skew analyses.
- Hoping to acquire the full data is often unrealistic.
- But, may be able to collect data selectively to enrich the incomplete network.
Issues to Consider

**Goal**

- Observe as many new nodes as possible
- Find triangles in the incomplete network
- ...

**Type of queries allowed**

- Can ask for and obtain:
  - all the edges of a node
  - a random edge of a node
  - the last $k$ communications made by a node
- ...

Issues to Consider

**Goal**

- Observe as many new nodes as possible
- Find triangles in the incomplete network
- …

**Type of queries allowed**

- Can ask for and obtain:
  - all the edges of a node
  - a random edge of a node
  - the last $k$ communications made by a node
- …
MaxOutProbe
Problem Definition

- **Given**
  - An incomplete network $\hat{G}$ that is part of a larger, unseen network $G$
  - A probing budget $b > 0$

- **Goal**
  - Select $b$ nodes from $\hat{G}$ that, when probed (in batch), bring as many new nodes as possible into $\hat{G}$

- **Assumption**
  - When a node is probed, **all** of its neighbors from $G$ are observed
Running Example: $\hat{G}$
Running Example

Which yellow nodes are adjacent to many green nodes?
Running Example: Which yellow nodes are adjacent to many green nodes?
MaxOutProbe: Outline

1. Using $\hat{G}$, estimate each node $u$’s true degree $d_u$ in $G$

2. Estimate the number of neighbors $u$ has inside $\hat{G}$
   - Using $\hat{G}$, estimate the average clustering coefficient $C$ of $G$

3. Using #1 and #2, estimate the number of neighbors $u$ has outside $\hat{G}$
MaxOutProbe (cont.)

\[ d_{u}^{out} = d_{u} - d_{u}^{in} = d_{u} - \left( d_{u}^{known} + d_{u}^{unknown} \right) \]
MaxOutProbe (cont.)

\[ d_u^{out} = d_u - d_u^{in} = d_u - (d_u^{known} + d_u^{unknown}) \]
Estimating Degrees $d_u$

- Calculate the average scaling factor $s$ such that on average, a node’s true degree can be approximated by $s$ times its observed degree.
- How do we calculate $s$?
  - Sample a small number of high degree nodes from $\hat{G}$
  - Observe the ratio of their true degrees to their observed degrees.
Estimating Internal Degrees

• Challenge
  • Given the structure of $\hat{G}$, how can we estimate the number of neighbors a node has inside $\hat{G}$?

• Observation: Nodes tend to cluster
  • If $u$ has many friends-of-friends inside $\hat{G}$, chances are $u$ is connected to some of them

• How many?
  • Use clustering coefficient to figure it out
  • Among the wedges, what fraction are closed triangles?
Running Example

- $u$ has 4 friends-of-friends (red-lined yellow circles)
- $C = \text{estimate of graph's clustering coefficient}$
- Estimate that $u$ is connected to $4C$ of these nodes
Estimating C

• Reuse nodes probed during degree-estimation step

• When probed, what fraction of their friends-of-friends were they connected to?
Unbiased Estimates

• **IF** we know that
  
  • $\hat{G}$ was produced by sampling nodes or edges uniformly at random from $G$, **AND**
  
  • the size of $G$

• **THEN** we can get unbiased estimates of $s$ and $C$

Experiments
## Datasets

<table>
<thead>
<tr>
<th>Network</th>
<th># of Nodes</th>
<th># of Edges</th>
<th>Transitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Retweets</td>
<td>40K</td>
<td>46K</td>
<td>0.03</td>
</tr>
<tr>
<td>Twitter Replies</td>
<td>261K</td>
<td>309K</td>
<td>0.002</td>
</tr>
<tr>
<td>Enron Emails</td>
<td>84K</td>
<td>326K</td>
<td>0.08</td>
</tr>
<tr>
<td>Yahoo! IM</td>
<td>100K</td>
<td>595K</td>
<td>0.08</td>
</tr>
<tr>
<td>Amazon Books</td>
<td>270K</td>
<td>741K</td>
<td>0.21</td>
</tr>
<tr>
<td>Youtube Videos</td>
<td>167K</td>
<td>1M</td>
<td>0.007</td>
</tr>
</tbody>
</table>
# Baseline & Competing Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HighDeg</td>
<td>Select nodes with the highest degree.</td>
</tr>
<tr>
<td>LowDeg</td>
<td>Select nodes with the lowest degree.</td>
</tr>
<tr>
<td>HighDisp</td>
<td>Select nodes with the highest dispersion.</td>
</tr>
<tr>
<td>LowDisp</td>
<td>Select nodes with the lowest dispersion.</td>
</tr>
<tr>
<td>CrossCom</td>
<td>Select nodes with the highest fraction of neighbors outside of their community (detected by Louvain Method).</td>
</tr>
<tr>
<td>HighCC</td>
<td>Select nodes with the highest clustering coefficients.</td>
</tr>
<tr>
<td>LowCC</td>
<td>Select nodes with the lowest clustering coefficients.</td>
</tr>
<tr>
<td>Random</td>
<td>Randomly select nodes from the sample.</td>
</tr>
</tbody>
</table>
Experimental Setup

• 20 trials
• Sample 10% of G’s edges using:
  • Random node sampling
  • Random edge sampling
  • Random walk
  • Random walk with jumps
• Run experiments at budgets $b$ in \{1\%, 2\%, 3\%, 4\%, 5\%\} of the # of nodes in each network
• Evaluate the quality of the enhanced graph by counting how many nodes it has
Results

- Compared to random probing, MaxOutProbe outperforms High Degree probing (the best baseline) by 4% - 36% on average
Results

- Small improvements are because of tiny clustering coefficients

Twitter Replies, C = 0.002, Random Walk Sample
Summary of MaxOutProbe

• Goal: Observe as many new nodes as possible
• Query: Returns all the edges of a node
• MaxOutProbe
  • Makes no assumptions about how the incomplete graph with generated or observed
  • Takes clustering coefficient into account
  • Improves performance over the best baseline algorithm (i.e., high-degree) by 4% to 36%
    • Improvement depends on G’s clustering coefficient
    • Tiny C, less improvement
Thank You!

• Funded by LLNL, NSF, DARPA, and DTRA

• Paper:

• Question to ssoundarajan@gmail.com