Eight Friends are Enough: Social Graph Approximation via Public Listings

Joseph Bonneau, Jonathan Anderson, Ross Anderson, Frank Stajano

University of Cambridge Computer Laboratory
Facebook Features & Privacy Backlashes

- News Feed (Sep 2006)
- Beacon (Nov 2007)
- “New Facebook” (Sep 2008)
- Terms of Use (Feb 2009)
- New Product Pages (Mar 2009)
Public Search Listings, Sep 2007
Public Search Listings

• Unprotected against crawling
• Indexed by search engines
• Opt out—but most users don't know it exists!
Entity Resolution
Utility

Promotion via Network Effects
“Your name, network names, and profile picture thumbnail will be available in search results across the Facebook network and those limited pieces of information may be made available to third party search engines. This is primarily so your friends can find you and send a friend request.”

-Facebook Privacy Policy
Legal Status

Much More Info Now Included...
One year ago today, the Fair Copyright for Canada Facebook group was launched. The past twelve months have been remarkable - thousands of Canadians have spoken out on copyright reform, with the issue capturing political and public attention as never before. While the issue is quiet politically at the moment (copyright reform was in the Speech from the Throne, but economic concerns are understandably taking priority), there is little doubt that it will return to the legislative agenda.

Group Type
This is an open group. Anyone can join and invite others to join.

Admins
- Michael
Obvious Attack

• Initially returned new friend set on refresh

• Can find all $n$ friends in $O(n \cdot \log n)$ queries
  • The Coupon Collector’s Problem
  • For 100 Friends, need 65 page refreshes

• As of Jan 2009, friends fixed per IP address
## Fun with Tor

### UK
- David Cottingham
- Eirik George Tsarpalis
- Emma Alden
- Luke Church
- Stella Nordhagen
- David J Hornsby
- Justin Palfreyman
- Jillian Sullivan

### Germany
- Shoshana Freisinger
- Lauren Duffey
- Conor Loftus-Sweetland
- Will Cordingley
- Srilakshmi Raj
- Sarita Kristina Sylvester
- Brian Brown
- Gary Champagne

### USA
- Melanie Kannokada
- Shoshana Freisinger
- Russ Heddeston
- Conor Loftus-Sweetland
- Gustav Rydstedt
- Seth Ort
- Cameron Lochte
- Ben Skolnik

### Australia
- Shoshana Freisinger
- Federico Baradello
- Lauren Duffey
- Adrian Boscolo-Hightower
- Justin David Carl
- Katie Gunderson
- Ankit Garg
- Srilakshmi Raj
Attack Scenario

- Spider all public listings
  - Our experiments crawled 250 k users daily
  - Implies ~800 CPU-days to recover all users

- Compute functions on sampled graph
Abstraction

- Take a graph $G = \langle V, E \rangle$

- Randomly select $k$ out-edges from each node

- Result is a sampled graph $G_k = \langle V, E_k \rangle$

- Try to approximate $f(G) \approx f_{\text{approx}}(G_k)$
Approximable Functions

- Node Degree
- Dominating Set
- Betweenness Centrality
- Path Length
- Community Structure
Experimental Data

• Crawled networks for Stanford, Harvard universities

• Representative sub-networks

<table>
<thead>
<tr>
<th></th>
<th># Users</th>
<th>Mean $d$</th>
<th>Median $d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford</td>
<td>15043</td>
<td>125</td>
<td>90</td>
</tr>
<tr>
<td>Harvard</td>
<td>18273</td>
<td>116</td>
<td>76</td>
</tr>
</tbody>
</table>
Stanford Histogram

Degree Distribution

- **data**
- **(1-1000): y = 0.7 * x ^ -1.27**

Proportion of Network

Degree

0.010

0.008

0.006

0.004

0.002

0.000

0

200

400

600

800

1000
Harvard Histogram

Degree Distribution

- Green dots: data
- Pink line: \( y = 0.92 \times x^{-1.34} \)
Comparison

Stanford

Harvard

Networks have very similar structure
Stanford Log-Log plot

Degree Distribution

- data
- (1-200): $y = 0.03 \times x^{-0.5}$
- (1-1000): $y = 0.7 \times x^{-1.27}$
- (200-1000): $y = 17697.9 \times x^{-2.98}$
Harvard Log-Log plot

**Degree Distribution**

- Data
- (1-200): $y = 0.04 \times x^{-0.58}$
- (1-1000): $y = 0.92 \times x^{-1.34}$
- (200-1000): $y = 8297.66 \times x^{-2.87}$
Take a graph $G = \langle V, E \rangle$

Randomly select $k$ out-edges from each node

Result is a sampled graph $G_k = \langle V, E_k \rangle$

Try to approximate $f(G) \approx f_{\text{approx}}(G_k)$
Estimating Degrees

- Convert sampled graph into a directed graph
  - Edges originate at the node where they were seen
- Learn exact degree for nodes with degree < $k$
  - Less than $k$ out-edges
- Get random sample for nodes with degree $\geq k$
  - Many have more than $k$ in-edges
Estimating Degrees

Average Degree: 3.5
Estimating Degrees

Sampled with $k=2$
Estimating Degrees

Degree known exactly for one node
Estimating Degrees

Naïve approach: Multiply in-degree by average degree / \( k \)
Raise estimates which are less than $k$
Estimating Degrees

Nodes with high-degree neighbors underestimated
Estimating Degrees

Iteratively scale by current estimate / k in each step
Estimating Degrees

After 1 iteration
Estimating Degrees

Normalise to estimated total degree
Convergence after $n > 10$ iterations
Estimating Degrees

- Converges fast, typically after 10 iterations
- Absolute error is high—38% average
  - Reduced to 23% for nodes with $d \geq 50$
- Still accurately can pick high degree nodes
Aggregate of $x$ highest-degree nodes
Comparison of sampling parameters
Dominating Sets

• Set of Nodes $D \subseteq V$ such that
  
  $$D \cup \text{Neighbours}(D) = V$$

• Set allows viewing the entire network

• Also useful for marketing, trend-setting
Trivial Algorithm: Select High-Degree Nodes in Order
In fact, finding minimal dominating set is NP-complete
Dominating Sets

Greedy Algorithm: select for maximal coverage
Dominating Sets

Greedy Algorithm: select for maximal coverage
Dominating Sets

Shown to perform adequately in practice
Works Well on Sampled Graph

Growth of Dominated Set

- Greedy, complete graph
- Greedy, k=8
- Degree only, complete graph
- Random, complete graph

% Network Coverage vs. Nodes Included
Insensitive to Sampling Parameter!

Surprising: Even $k = 1$ performs quite well
Shortest Paths

- Social networks shown to be “small world”
- Short paths should exist, even for large graphs
- Short paths can be used for social engineering
Floyd-Warshall Algorithm

- Finds shortest distance between all pairs of nodes
- Dynamic programming – $O(V^3)$ over $V^2$ nodes
- Think Dijkstra, but for all vertices
Floyd-Warshall Algorithm
Floyd-Warshall Algorithm

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>∞</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>∞</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>∞</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Floyd-Warshall Algorithm
Short Paths Still Exist in Sampled Graph

Reachable Nodes vs. Path Length

- Complete
- \(k=100\)
- \(k=50\)
- \(k=10\)
- \(k=8\)
- \(k=5\)
- \(k=2\)
Centrality

- A measure of a node's importance

- *Betweenness centrality*:

\[
C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}
\]

- Measures the shortest paths in the graph that a particular vertex is part of
Centrality

$C_B(v_7) = ?$
Centrality

\[ C_B(v_7) = \frac{0}{1} + \]
Centrality

\[ C_B(v_7) = \frac{0}{1} + \frac{0}{2} + \]

Diagram: A network of nodes representing centrality.
$C_B(v_7) = \frac{0}{1} + \frac{0}{2} + \frac{4}{4}$
Message Interception Scenario

- Messages sent via shortest (least-cost) paths
- Adversary can compromise $x$ nodes
- How much traffic can s/he intercept?

$$ P_{\text{intercept}}(v_s, v_d) = \frac{C_B(v)}{|V|^2} $$
Message Interception

![Message Interception Probability Graph](image)

- Maximum Centrality
- Max Centrality (k=10)
- Max Centrality (k=8)
- Max Centrality (k=5)
- Max Centrality (k=2)
- Max Centrality (k=1)
- Random Selection

Probability of Intercept vs. Compromised Nodes
Community Detection

- Goal: Find highly-connected sub-groups
- Measure success by high modularity:

\[ Q = \frac{1}{2m} \sum_{v,w} \left[ A_{vw} - \frac{d(v)d(w)}{2m} \right] \]

- Ratio of intra-community edges to random
- Normalised to be between -1 and 1
Community Detection

- Clausen et. al 2004 – find maximal modularity in $O(n \lg^2 n)$
- Track marginal modularity, update neighbours on each merge
Community Detection

Q=0.04
Community Detection

Q = 0.08

Graph with nodes 1, 2, 3, 4, and edges with weights 0.06, 0.03, 0.0125, 0.025, 0.035, 0.035.
Community Detection

Q=0.14
Community Detection

Q=0.175
Community Detection

Q=0.2125
Community Detection

Q=0.2225
Community Detection

![Graph showing community detection with varying numbers of communities and modularity values for different values of k.]
Conclusions

• Social graph is fragile to partial disclosure
  • Consistent with Danezis/Wittneben, Nagaraja results

• Public Listings Leak Too Much
  • Dominating sets, centrality, communities in particular

• SNS operators need a dedicated privacy review team
  • Comparable to security audit & penetration testing
Questions?

jcb82@cl.cam.ac.uk

jra40@cl.cam.ac.uk