Analyzing Private Network Data Using Output Perturbation

Gerome Miklau

Joint work with

Michael Hay, David Jensen, Don Towsley
University of Massachusetts, Amherst

Vibhor Rastogi, Dan Suciu
University of Washington

September 11, 2009
Social network analysis -- transformed

1977

“Zachary’s Karate Club”

W. W. Zachary
An information flow model for conflict and fission in small groups
Journal of Anthropological Research

34 nodes
78 edges

2008

Global IM Network

J. Leskovec and E. Horvitz.
Planetary-scale views on a large instant-messaging network
Conference on the World Wide Web

180 million nodes
1.3 billion edges
The opportunity is not being realized

Major obstacle to data dissemination: privacy risk.

• Common outcomes:
  • No availability
  • Limited availability
    • only within institutions who own the data, or a among limited set of researchers who have negotiated access.
  • Availability, at a cost
    • Privacy of participants may be violated, bias or inaccuracy in released data.
Analysis of private networks

Can we permit analysts to study networks without revealing sensitive information about participants?

Example analyses based on network topology:

• Properties of the degree distribution
• Motif analysis
• Community structure
• Processes on networks: routing, rumors, infection
• Resiliency
Synthetic data release

Original network

Anonymization

Naive anonymization
Synthetic data release

Original network

Naive anonymization

- Create topological similarity [Liu, SIGMOD 08] [Zhou, ICDE 08] [Zou, VLDB 09]
Synthetic data release

• Create topological similarity [Liu, SIGMOD 08] [Zhou, ICDE 08] [Zou, VLDB 09]

• Randomize edges [Rastogi, VLDB 07] [Ying, SDM 2008]
Synthetic data release

DATA OWNER

Original network

ANALYST

Anonymization

Node clustering

• Create topological similarity [Liu, SIGMOD 08] [Zhou, ICDE 08] [Zou, VLDB 09]

• Randomize edges [Rastogi, VLDB 07] [Ying, SDM 2008]

• Clustering/summarization [Campan, PinKDD 08] [Hay, VLDB 08] [Cormode, VLDB 08] [Cormode, VLDB 09]
Dwork, McSherry, Nissim, Smith [Dwork, TCC 06] have described an output perturbation mechanism satisfying differential privacy.

Comparatively few results for these techniques applied to graphs.
Synthetic data release v. output perturbation

- Synthetic data release
  
- Output perturbation
  
- Model-based synthetic data

<table>
<thead>
<tr>
<th></th>
<th>Usability</th>
<th>Privacy</th>
<th>Accuracy</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic data release</td>
<td>good</td>
<td>no formal guarantees</td>
<td>no formal guarantees</td>
<td>sometimes bad</td>
</tr>
<tr>
<td>Output perturbation</td>
<td>bad for practical analyses</td>
<td>formal guarantees</td>
<td>provable bounds</td>
<td>very good</td>
</tr>
<tr>
<td>Model-based synthetic data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The best of both worlds ??
Outline of the talk

• Differential privacy defined, adapted to graphs

• The accuracy of two topological analyses under differential privacy:
  • Motif analysis
  • Degree distribution

• Future goals and open questions
The differential guarantee

Two graphs are neighbors if they differ by at most one edge.
Differential privacy

A randomized algorithm $A$ provides $\epsilon$-differential privacy if:
for all neighboring graphs $G$ and $G'$, and
for any set of outputs $S$:

$$
Pr[A(G) \in S] \leq e^\epsilon Pr[A(G') \in S]
$$

Epsilon is usually small: e.g. if $\epsilon = 0.1$ then $e^\epsilon \approx 1.10$

$\downarrow$ epsilon = $\uparrow$ stronger privacy
Calibrating noise

• The following algorithm for answering Q is \( \varepsilon \)-differentially private:

\[
A 
\xrightarrow{Q} Q(G) + \text{Laplace}(\Delta Q / \varepsilon)
\]

\(\Delta Q\): The maximum change in Q, over any two neighboring graphs
Examples of query sensitivity

<table>
<thead>
<tr>
<th>query</th>
<th>sensitivity</th>
<th>truth</th>
<th>noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>$\Delta Q$</td>
<td>$Q(G)$</td>
<td>$\text{Lap}(\Delta Q / \varepsilon)$</td>
</tr>
<tr>
<td>$\deg_A$ (degree of node A)</td>
<td>1</td>
<td>$\deg_{	ext{Dave}}(G) = 4$</td>
<td>4+$\text{Lap}(2)$</td>
</tr>
<tr>
<td>$\text{cnt}_i$ (# nodes with degree i)</td>
<td>2</td>
<td>$\text{cnt}_4(G) = 4$</td>
<td>4+$\text{Lap}(4)$</td>
</tr>
<tr>
<td>$D = [\deg_{\text{Dave}}, \deg_{\text{Ed}}, \deg_{\text{Fred}}]$</td>
<td>2</td>
<td>$D(G) = [4, 4, 2]$</td>
<td>$[4 + \text{Lap}(4), 4 + \text{Lap}(4), 2 + \text{Lap}(4)]$</td>
</tr>
</tbody>
</table>

\[
\text{deg}_A \text{ (degree of node A)}
\]

\[
\text{cnt}_i \text{ (number of nodes with degree } i\text{)}
\]

\[
D = [\text{deg}_{\text{Dave}}, \text{deg}_{\text{Ed}}, \text{deg}_{\text{Fred}}]
\]

Sunday, September 13, 2009
Differential privacy for networks

A participant’s sensitive information is not a single edge.

- **edge $\epsilon$-differential privacy**: algorithm output is largely indistinguishable whether or not any single edge is present or absent.

- **$k$-edge $\epsilon$-differential privacy**: algorithm output is largely indistinguishable whether or not any set of $k$ edges is present or absent.

- **node $\epsilon$-differential privacy**: algorithm output is largely indistinguishable whether or not any single node (and all its edges) is present or absent.

Suppose $\Delta Q=1$. Then Laplace(100) satisfies:

1-edge 0.01-differential privacy
10-edge 0.1-differential privacy
Accurate motif analysis is hard

- Motif analysis measures the frequency of occurrence of small subgraphs in a network.

- Common example: *transitivity* in the network:
  - when A is friends with B and C, are B and C also friends?
  - $Q_{\text{TRIANGLE}}$: return the number of triangles in the graph

$$Q_{\text{TRIANGLE}}(G) = 0 \quad Q_{\text{TRIANGLE}}(G') = n-2$$

High Sensitivity:
$$\Delta Q_{\text{TRIANGLE}} = O(n)$$

For improved accuracy, see:
[Rastogi, PODS 09]
[Nissim, STOC 07]
Accurate degree sequence estimation is possible

- Degree sequence: the list of degrees of each node in a graph.
- A widely studied property of networks.

\[ [1,1,2,2,4,4,4,4] \]

degree sequence of \( G \)

Orkut crawl

Inverse cumulative distribution
Alternative queries for the degree sequence

<table>
<thead>
<tr>
<th>count queries</th>
<th>vector queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{deg}_A</td>
<td>\text{D} [\text{deg}_A, \text{deg}_B, \ldots]</td>
</tr>
<tr>
<td>\text{rnk}_i</td>
<td>\text{S} [\text{rnk}_1, \text{rnk}_2, \ldots \text{cnt}_n]</td>
</tr>
</tbody>
</table>

\text{deg}_A: return degree of node A

\text{rnk}_i: return the rank \(i^{th}\) degree

\[
\begin{array}{c}
G \\
\begin{array}{c}
\text{Alice} \quad \text{Bob} \quad \text{Carol} \\
\text{Dave} \quad \text{Ed} \\
\text{Fred} \quad \text{Greg} \quad \text{Harry}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
G' \\
\begin{array}{c}
\text{Alice} \quad \text{Bob} \quad \text{Carol} \\
\text{Dave} \quad \text{Ed} \\
\text{Fred} \quad \text{Greg} \quad \text{Harry}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\text{D}(G) = [1,4,1,4,4,2,4,2] \\
\text{S}(G) = [1,1,2,2,4,4,4,4]
\end{array}
\]

\[
\begin{array}{c}
\text{D}(G') = [1,4,1,3,3,2,4,2] \\
\text{S}(G') = [1,1,2,2,3,3,4,4]
\end{array}
\]

\[
\begin{array}{c}
\Delta \text{D}=2 \\
\Delta \text{S}=2
\end{array}
\]
Using the sort constraint

- The output of the sorted degree query is not (in general) sorted.

- We derive a new sequence by computing the closest non-decreasing sequence: i.e. minimizing L2 distance.
Experimental results

powerlaw, $\alpha=1.5$, $n=5M$

(100-edge, 0.1-differential privacy) $\varepsilon=.001$

(10-edge, 0.1-differential privacy) $\varepsilon=.01$
Experimental results, continued

<table>
<thead>
<tr>
<th>Network</th>
<th>Parameters</th>
<th>Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>livejournal</td>
<td>n=5.3M</td>
<td><img src="image1.png" alt="Graph" /></td>
</tr>
<tr>
<td>orkut</td>
<td>n=3.1M</td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
<tr>
<td>powerlaw</td>
<td>α=1.5, n=5M</td>
<td><img src="image3.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

For ε=.001:

- livejournal: ![Graph](image1.png)
- orkut: ![Graph](image2.png)
- powerlaw: ![Graph](image3.png)

For ε=.01:

- livejournal: ![Graph](image1.png)
- orkut: ![Graph](image2.png)
- powerlaw: ![Graph](image3.png)
Inference does not weaken privacy

1. Formulate $S$, having constraints $\gamma_s$
2. Submit $S$
3. Perform inference $\tilde{S}$

DATA OWNER

ANALYST

$S(G) +$ noise

$\gamma_s$

Inference

$\tilde{S}$
Accuracy is improved without sacrificing privacy!

• Improved accuracy is possible because standard Laplace noise is sufficient (but not necessary) for differential privacy.

• By using constraints and inference, we effectively apply a different noise distribution -- more noise where it is needed, less otherwise.

• An immediate consequence: the accuracy achieved depends on the input sequence.

\[
\text{Mean squared error, } \tilde{S} \quad \Theta(n/\epsilon^2)
\]

\[
\text{Mean squared error, } \bar{S} \quad O(d\log^3 n/\epsilon^2)
\]
Toward differentially-private synthetic data

- To realize the benefits of synthetic data, data owner can release noisy parameters of network model.

- Baseline: the degree distribution as network model
  - Deriving the power law parameter: very accurate
  - Measuring clustering coefficient: not constrained by deg. distr.
A few open questions

• For graph measure X, how accurately can we estimate X under differential privacy?

• Is it socially acceptable to offer weaker privacy protection to high-degree nodes (as in k-edge differential privacy)?

• Can we generate accurate synthetic networks under differential privacy?
  
  • Can the analyst explore an approximate synthetic network, then query the original network under output perturbation?
Questions?

Additional details on our work may be found here:


References


References (con’t)


THEOREM 1. Denote $L_k = \min_{j \in [k, n]} \max_{i \in [1, j]} M[i, j]$ and $U_k = \max_{i \in [1, k]} \min_{j \in [i, n]} M[i, j]$. The minimum $L_2$ solution $\bar{s} = \text{MIN}_2(\tilde{s}, \gamma s)$, is unique and given by: $\bar{s}[k] = L_k = U_k$. 