Preserving the Privacy in Online Social Networks Using Enhanced Clustering Algorithms

Rumman Ahmed
Outline

- Introduction
- Related Work
- Problem Statement
- Experimental Results
- Conclusion
Introduction

- The role of social networks for the modern population
- Public and private data sharing by users
- Collection of data from social networks
- Protecting the anonymity of a social network
### Privacy Settings and Tools

#### Who can see my stuff?
- **Who can see your future posts?**
  - Only Me
  - [Edit]
- **Review all your posts and things you're tagged in**
  - [Use Activity Log]
- **Limit the audience for posts you've shared with friends of friends or Public?**
  - [Limit Past Posts]

#### Who can contact me?
- **Who can send you friend requests?**
  - Friends of friends
  - [Edit]

#### Who can look me up?
- **Who can look you up using the email address you provided?**
  - Friends of friends
  - [Edit]
- **Who can look you up using the phone number you provided?**
  - Friends of friends
  - [Edit]
- **Do you want search engines outside of Facebook to link to your Profile?**
  - No
  - [Edit]
Introduction

- The role of social networks for the modern population
- Public and private data sharing by users
- Collection of data from social networks
- Protecting the anonymity of a social network
Related Work

- Sequentially clustering to anonymize a network
- The k-member clustering problem
- Data and structural k-anonymity in social networks
Problem Statement

- A social network can be represented as a graph
- Clustering a social network...the issue at hand

Figure 1. A social network as a graph
Problem Statement

- Preventing single user clusters
- Keeping information loss at a minimum
- Increasing degree of anonymity

Figure 1. A social network as a graph
Problem Statement - The Algorithm in Use

Algorithm 1: Generating Equal-Sized Clusters

1. Determine cluster size
2. Calculate initial clusters with built-in original clustering algorithm
3. Sort points by the difference of their nearest cluster to the farthest cluster
4. Assign points to nearest cluster until the cluster size is reached. Continue with all points
5. Compute current cluster means
6. For each point, compute the distances to the cluster means
7. Sort points based on the difference between the current assignment and the best possible alternate assignment.
8. For each point by priority:
   a. For each other cluster:
      i. If there is an element wanting to leave the other cluster, swap the two elements if it's an improvement, without violating cluster max size
   b. If the element was not changed, add to outgoing transfer list.
9. If no more transfers were done (or max iteration threshold was reached),

Algorithm 1: Generating Equal-Sized Clusters

1. Determine cluster size
2. Calculate initial clusters with built-in original clustering algorithm
3. Sort points by the difference of their nearest cluster to the farthest cluster
4. Assign points to nearest cluster until the cluster size is reached. Continue with all points
5. Compute current cluster means
6. For each point, compute the distances to the cluster means
7. Sort points based on the difference between the current assignment and the best possible alternate assignment.
8. For each point by priority:
   a. For each other cluster:
      i. If there is an element wanting to leave the other cluster, swap the two elements if it's an improvement, without violating cluster max size
   b. If the element was not changed, add to outgoing transfer list.
9. If no more transfers were done (or max iteration threshold was reached),

Experimental Results

- Traditional clustering algorithms compared to our enhanced algorithms
- Metrics
  - Information Loss
  - Degree of Anonymization
  - Running Time
Experimental Results

- Experiment Details
  - Dataset: 5000 Yelp users
  - Enhanced versions of K-Means, Mean-Shift, and Affinity Propagation
  - Implemented in Python
### Experimental Results - Tables

#### Table 1: K-Means, 5000 users

<table>
<thead>
<tr>
<th>K (number of clusters)</th>
<th>Traditional Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
<th>Enhanced Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.64%</td>
<td>0.0016</td>
<td>5</td>
<td>0.82%</td>
<td>0.001</td>
<td>215</td>
</tr>
<tr>
<td>10</td>
<td>0.43%</td>
<td>0.0004</td>
<td>6</td>
<td>0.62%</td>
<td>0.002</td>
<td>152</td>
</tr>
<tr>
<td>25</td>
<td>0.27%</td>
<td>0.0002</td>
<td>6</td>
<td>0.47%</td>
<td>0.004</td>
<td>725</td>
</tr>
<tr>
<td>50</td>
<td>0.17%</td>
<td>0.0002</td>
<td>8</td>
<td>0.33%</td>
<td>0.01</td>
<td>711</td>
</tr>
<tr>
<td>100</td>
<td>0.12%</td>
<td>0.0002</td>
<td>10</td>
<td>0.3%</td>
<td>0.02</td>
<td>950</td>
</tr>
</tbody>
</table>

#### Table 2: Mean-Shift, 5000 users

<table>
<thead>
<tr>
<th>K (number of clusters)</th>
<th>Traditional Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
<th>Enhanced Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 (q 0.9)</td>
<td>1.1%</td>
<td>0.0002</td>
<td>35</td>
<td>0.55%</td>
<td>0.003</td>
<td>366</td>
</tr>
<tr>
<td>25 (q 0.5)</td>
<td>0.82%</td>
<td>0.0002</td>
<td>50</td>
<td>0.62%</td>
<td>0.005</td>
<td>554</td>
</tr>
<tr>
<td>35</td>
<td>0.66%</td>
<td>0.0002</td>
<td>67</td>
<td>0.73%</td>
<td>0.007</td>
<td>777</td>
</tr>
<tr>
<td>49 (q 0.2)</td>
<td>0.52%</td>
<td>0.0002</td>
<td>78</td>
<td>0.64%</td>
<td>0.01</td>
<td>772</td>
</tr>
<tr>
<td>140 (q 0.05)</td>
<td>0.29%</td>
<td>0.0002</td>
<td>104</td>
<td>0.65%</td>
<td>0.03</td>
<td>1056</td>
</tr>
</tbody>
</table>

#### Table 3: Affinity Propagation, 5000 users

<table>
<thead>
<tr>
<th>K (number of clusters)</th>
<th>Traditional Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
<th>Enhanced Information Loss</th>
<th>Degree of Anonymization</th>
<th>Running Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 (d 0.99)</td>
<td>0.75%</td>
<td>0.0002</td>
<td>9</td>
<td>1.0%</td>
<td>0.02</td>
<td>112</td>
</tr>
<tr>
<td>57 (d 0.9)</td>
<td>0.25%</td>
<td>0.0002</td>
<td>13</td>
<td>0.85%</td>
<td>0.04</td>
<td>156</td>
</tr>
<tr>
<td>65 (d 0.85)</td>
<td>0.26%</td>
<td>0.0002</td>
<td>13</td>
<td>0.78%</td>
<td>0.05</td>
<td>158</td>
</tr>
<tr>
<td>116 (d 0.6)</td>
<td>0.25%</td>
<td>0.0002</td>
<td>15</td>
<td>0.42%</td>
<td>0.00</td>
<td>245</td>
</tr>
<tr>
<td>131 (d 0.5)</td>
<td>0.26%</td>
<td>0.0002</td>
<td>16</td>
<td>0.38%</td>
<td>0.1</td>
<td>263</td>
</tr>
</tbody>
</table>
Experimental Results - Graphs

Enhanced Algorithms
Number of Clusters vs. Information Loss

Enhanced Algorithms
Number of Clusters vs. Degree of Anonymization

Enhanced K-Means

Information Loss vs. Degree of Anonymization
Result Analysis

- Information loss kept at a minimum
- Increase of the degree of anonymization
- Running time comparisons, enhanced algorithms impractical for large data
Conclusion

- Importance of privacy and anonymity in modern time
- Current solution of clustering has a drawback
- New proposed solution shows promising results and a stepping stone for the anonymity problem


Questions?