Identity Anonymization on Graphs

Kundan Singh [S4723435]

Background

- IBM Research paper by Liu & Terzi
- Privacy concern on Individual Network data
- specific graph-anonymization problem

Why K-Degree anonymization?

- ❑ **Only removing Identity of nodes doesn’t always guarantee Privacy**

- ✓ **Adversaries** can infer the identity of the nodes by solving a set of restricted isomorphism problems based on the **uniqueness of small random subgraphs** embedded in network.

- ❑ **Structure or basic degree of nodes can help to reveal identities of individuals.**

- ✓ structural similarity of the nodes in the graph determines the extent to which an individual in the network can be distinguished from others.

The Problem

To Create **K-degree** anonymized **Graph Ga:**

Given a **graph G** and an integer **k**,

- ❑ *Modify **Graph G** via a set of **edge-addition** or **deletion***
- ❑ *Every **node v** has the **same degree** with at least **k-1** other nodes.*

- ❑ ***additional requirement** that the minimum number of such edge-modifications is made:*
- ✓ *Preserve the utility of the original graph, while at the same time satisfy the **degree-anonymity constraint***

Problem Definition

Given a graph G(V,E) and an integer k :

find a **k-degree** anonymous graph **Gb(V, Eb**) with Eb ¥ E = E such that **Ga(G, Gb)** is minimized.

V *is a set of* ***nodes*** *and* ***E*** *the set of* ***edges*** *in G and* ***dG*** *to denote the* ***degree sequence of G***

Ga(G, Gb): *graph-anonymization cost*

Problem solving Approach

Two Step approach :

- **Degree Anonymization**

*Given the **degree** sequence **d** of the original input **graph G(V,E),** the algorithms output **a k-anonymous** degree sequence **db** such that the degree anonymization cost **Da** is minimized*

- **Graph Construction**

*Given the original **graph G(V,E)** and the desired **k-anonymous degree** sequence **db** output by the DP (or Greedy) algorithm, we construct a **k-degree** anonymous graph **Gb(V, Eb)** with **Eb ¥ E = E** and degree sequence **dGb** with **dGb = db**.*

Dataset

This network represents the "core" of the email-EuAll network, which also contains links between members of the institution and people outside of the institution.

Dataset statistics	
Nodes	1005
Edges	25571
Nodes in largest WCC	986 (0.981)
Edges in largest WCC	25552 (0.999)
Nodes in largest SCC	803 (0.799)
Edges in largest SCC	24729 (0.967)
Average clustering coefficient	0.3994
Number of triangles	105461
Fraction of closed triangles	0.1085
Diameter (longest shortest path)	7
90-percentile effective diameter	2.9

https://snap.stanford.edu/data/email-Eu-core.html

Degree Anonymization

- **Dynamic Programming algorithm**
- Greedy algorithm

Input : sorted degree sequence **d** of graph **G**

- Anonymization cost **C** is calculated
- To improve speed **O(n2) → O(*nk)***

Any group >= 2k-1 can broken into two subgroups with equal or lower overall degree-anonymization cost.

considering **t's** in the range **max{k,i-2k+1}**

recursion

$$\mathrm{D_A}\left(\mathbf{d}[1,i]\right) = \min_{\max\{k,i-2k+1\}\leq t\leq i-k}\left\{\mathrm{D_A}\left(\mathbf{d}[1,t]\right) + I\left(\mathbf{d}[t+1,i]\right)\right\}$$

Graph Construction

- ❑ **Construct Graph algorithm**
- ❑ Relaxed Graph algorithm
- ❑ Greedy Swap algorithm
- ❑ Priority algorithm

Input: degree sequence **d** of length **n**

- Check realizable:
  if sum is odd: Halt and return "No"

While **true**:
 if **d(i) < 0** then Halt and return "No"
 if sequence d are all zeros : Halt and return G(V,E)

Pick a **random node** v with d(v) > 0
Set d(v)=0

iterate over degree-sorted vertices
 add edges that for both available or not available the original graph as well

Evaluation & comparison

- **Anonymization cost**

  anonymization cost is very close to the Baseline cost also , the degree sequences that are solutions to the Degree Anonymization

- **Clustering coefficient**

  CC is almost equal of the original graph. Both negligible increments and decrements are observed.

- **Average Path length**

  anonymization process decreases the average path length of the output graph since new connections are added.

- **Edge Intersection**

  around 56% of edge intersection is obtained since we added the edge present in the original graph while graph construction.

Extension

- **SIMULTANEOUS EDGE ADDITIONS AND DELETIONS**

algorithm implicitly allows for both edge-additions and edge-deletions.

Conclusion

- **Difficult to model capability of attacker**

  Any topological structure of the graph can be potentially used to derive private information

- **Difficult to measure utility of graph**

  Not aware of any effective metrics to quantify the information loss incurred by the changes of its nodes and edges.

THANK YOU