Anonymizing Social Network

Data Protection & Privacy

Iman Emrani & Faramarz Zareian

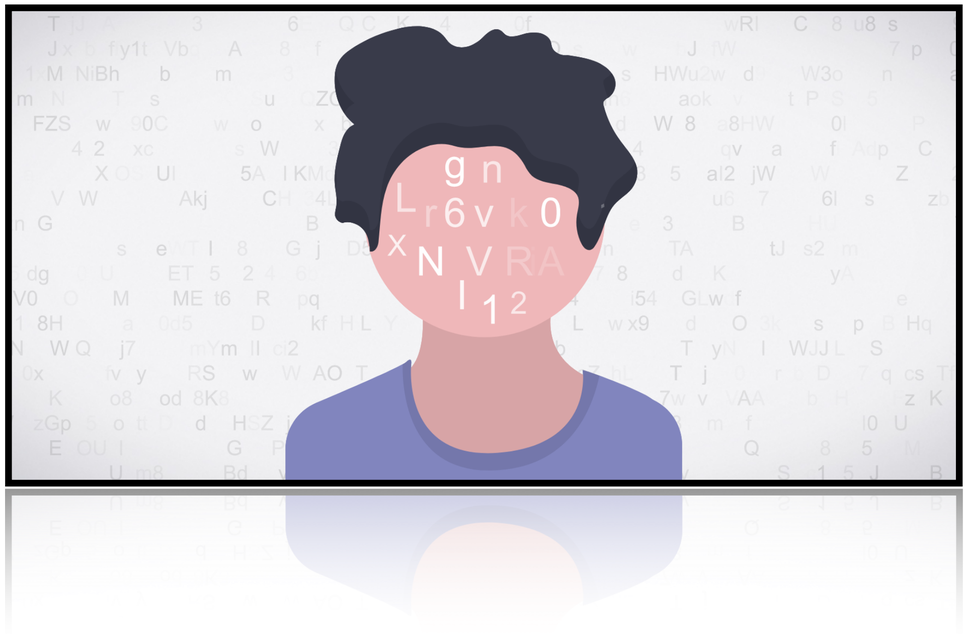
Social Network

- A social network describes entities and connections between them.

- The entities are often individuals; they are connected by personal relationships, interactions, or flows of information

Anonymization

It is the process of removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous.



The goal of the project

- To enable the useful analysis of social network data while protecting the privacy of individuals.

Main definitions

- **Data trustee** is a version of the data that permits useful analysis while protecting the privacy of the entities represented

- **Adversary** whose goal is to re-identify a known individual in the anonymized graph

- **Quasi-identifiers** are pieces of information that are not of themselves unique identifiers, but are sufficiently well correlated with an entity that they can be combined with other quasi-identifiers to create a unique identifier

- **synthetic identifiers** In order to preserve node identity in the graph of relationships, synthetic identifiers are introduced to replace names

Main definitions

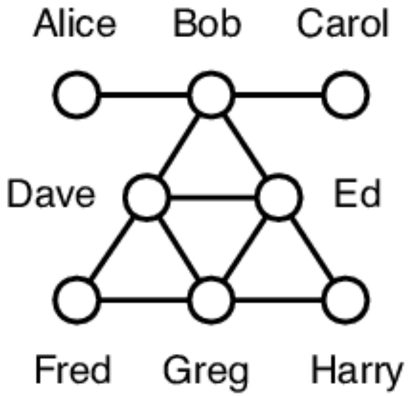
- **Isomorphic Graph**

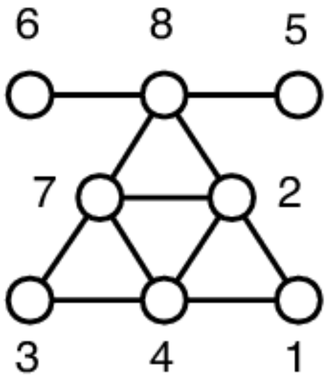
Definition 1 (Naive Anonymization) *The naive anonymization of a graph* $G = (V, E)$ *is an isomorphic graph,* $G_{na} = (V_{na}, E_{na})$*, defined by a random bijection* $f : V \rightarrow V_{na}$*. The edges of* G_{na} *are* $E_{na} = \{(f(x), f(x'))|(x, x') \in E\}$.

Privacy through the naive anonymization

Naive anonymization

Simplest method that simply removes all the identifier attributes before publishing the actual network. Furthermore, in order to maintain the node identity in the network, synthetic identifiers are generated and replaced by the original ones





Alice	6
Bob	8
Carol	5
Dave	7
Ed	2
Fred	3
Greg	4
Harry	1

Is it enough?

- Answer: NO

Synthetic identifiers reveal nothing about which node in the graph may be Bob. But by collecting information from external sources about an individual's relationships, an adversary may be able to re-identify individuals in the graph.

Adversary Knowledge

Vertex refinement queries

- We define a class of queries, which report on the local structure of the graph around a node.

- These queries are inspired by iterative vertex refinement.

- A technique originally developed to efficiently test for the existence of graph isomorphisms.

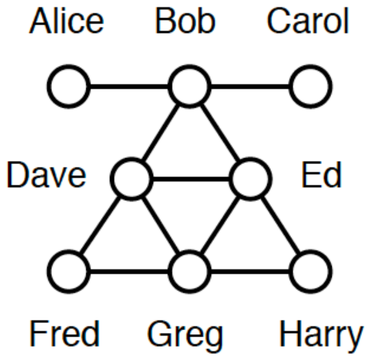
Vertex refinement queries

H0, simply returns the label of the nodes

H1(x) returns the degree of x,

H2(x) returns the list of each neighbors' degree

*Iterative computation of Hi(x) continues until no new vertices are distinguished



(a) graph

Node ID	$\mathcal{H}_0$	$\mathcal{H}_1$	$\mathcal{H}_2$
Alice	ϵ	1	$\{4\}$
Bob	ϵ	4	$\{1, 1, 4, 4\}$
Carol	ϵ	1	$\{4\}$
Dave	ϵ	4	$\{2, 4, 4, 4\}$
Ed	ϵ	4	$\{2, 4, 4, 4\}$
Fred	ϵ	2	$\{4, 4\}$
Greg	ϵ	4	$\{2, 2, 4, 4\}$
Harry	ϵ	2	$\{4, 4\}$

(b) vertex refinements

Subgraph knowledge queries

Subgraph knowledge queries

We measure the descriptive power of a query by counting the number of edges in the described subgraph; we refer to these as edge facts.

Figure 3 illustrates three subgraphs centered around Bob. The first simply asserts that Bob has (at least) three distinct neighbors, the second describes a tree of nodes near Bob, and the third relates nearby nodes in a subgraph. These informal query patterns use 3, 4 and 4 edge facts, respectively.

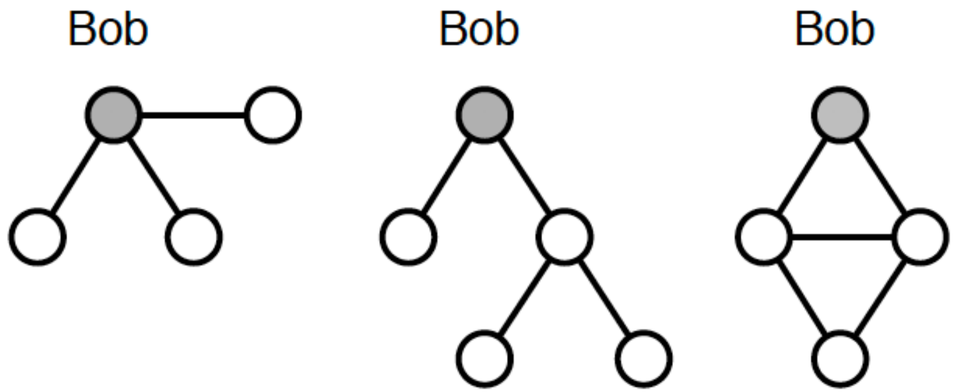


Figure 3

Graph Perturbation

It is applied to the graph after it has been naively anonymized (G_{na})

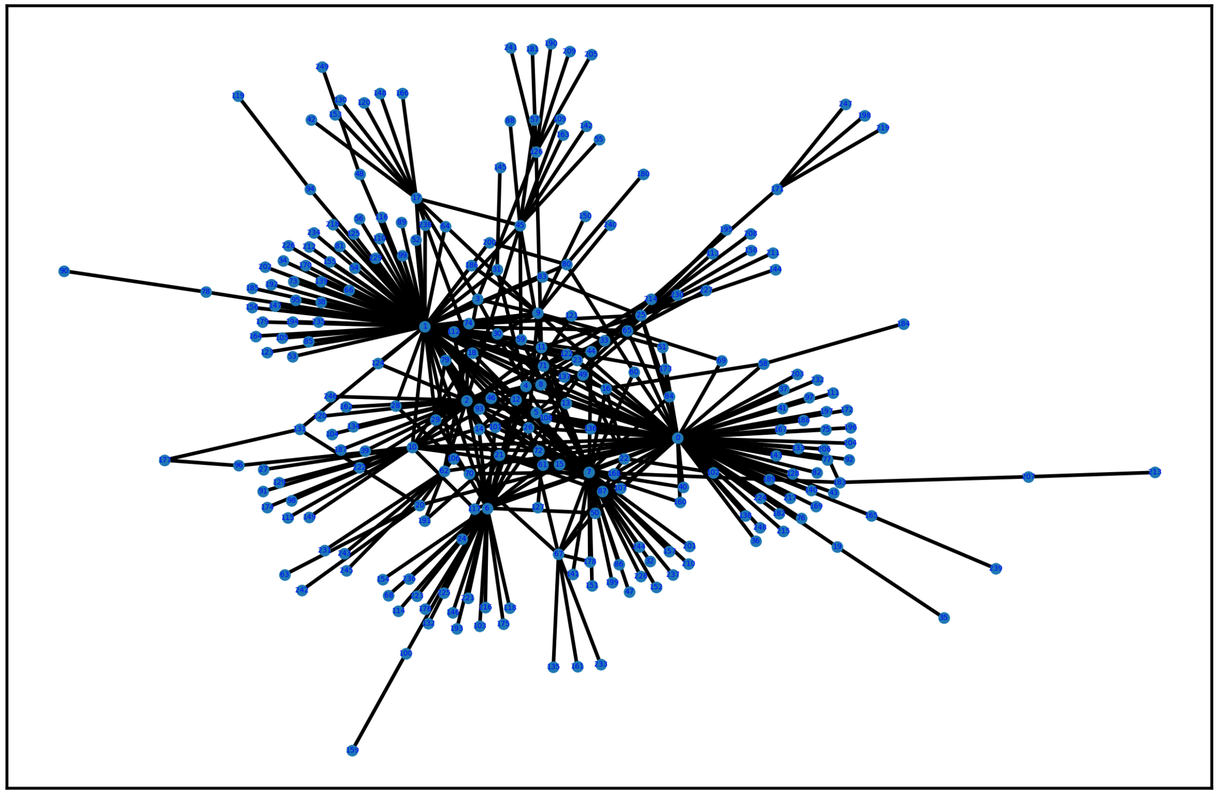
- The new graph $G_p = (V_p, E_p)$ is constructed from G_{na} through a sequence of m edge deletions followed by m edge insertions.
- Deletions are chosen uniformly at random from the set of all existent edges in G_{na}
- Insertions are chosen uniformly at random from the set of all non-existent edges of the interim graph.
- The nodes are unchanged

Python Implementation

Used Graph

Scale-Free Network Graph

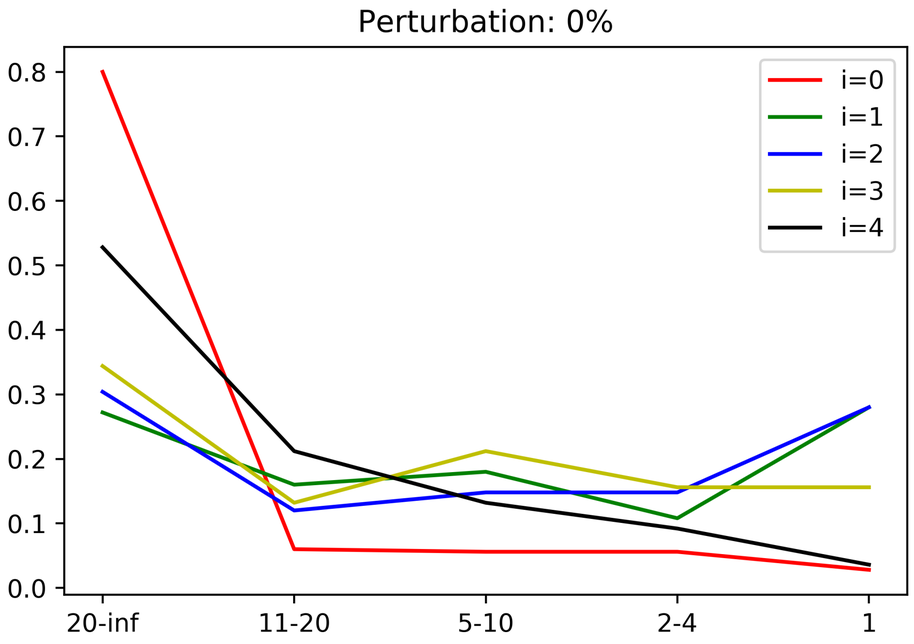
- We used Scale-Free Network Graph for our experiments
- It includes 250 nodes
- Structurally this kind of graph is similar to social network

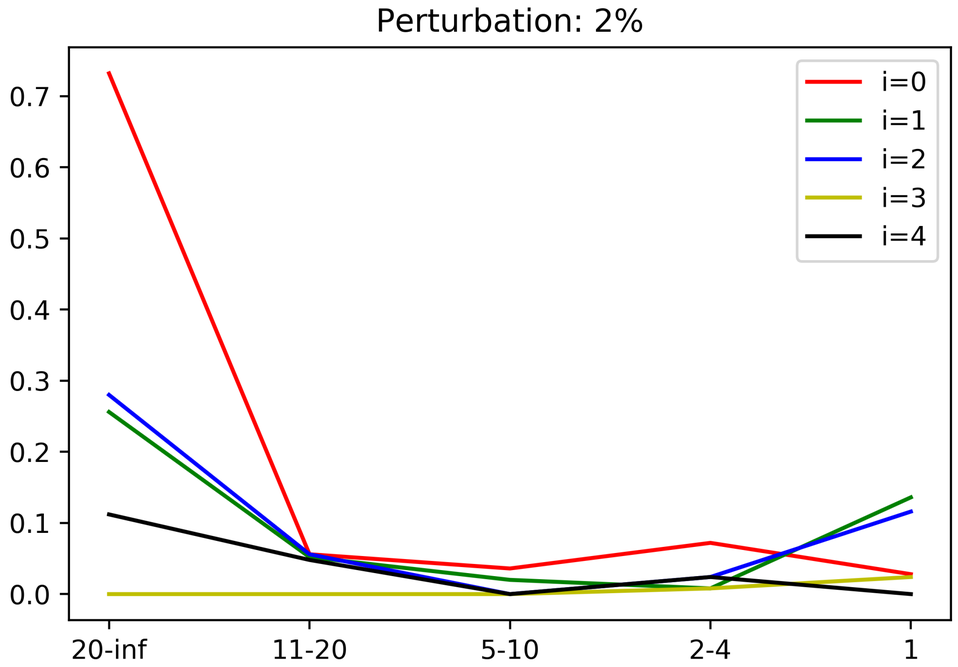


De-anonymization

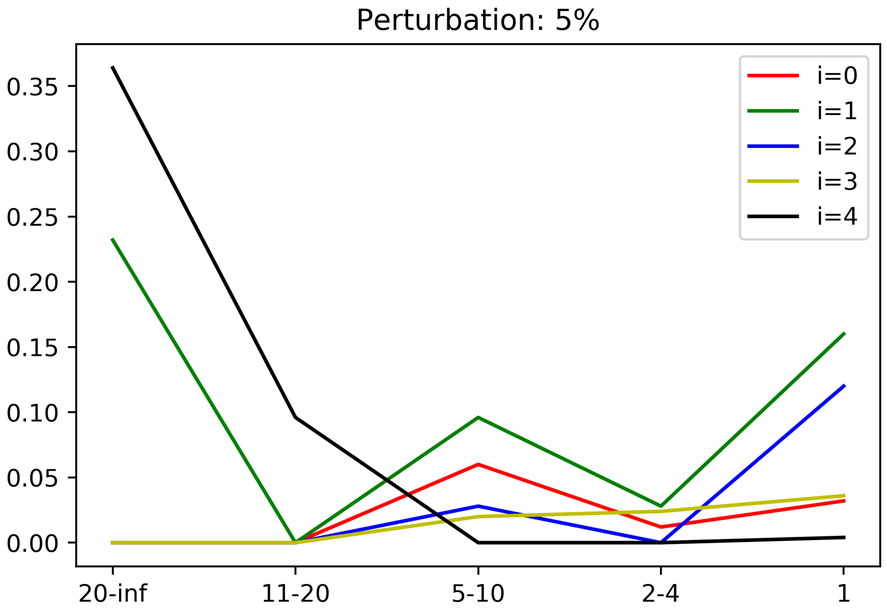
Vertex refinement technique

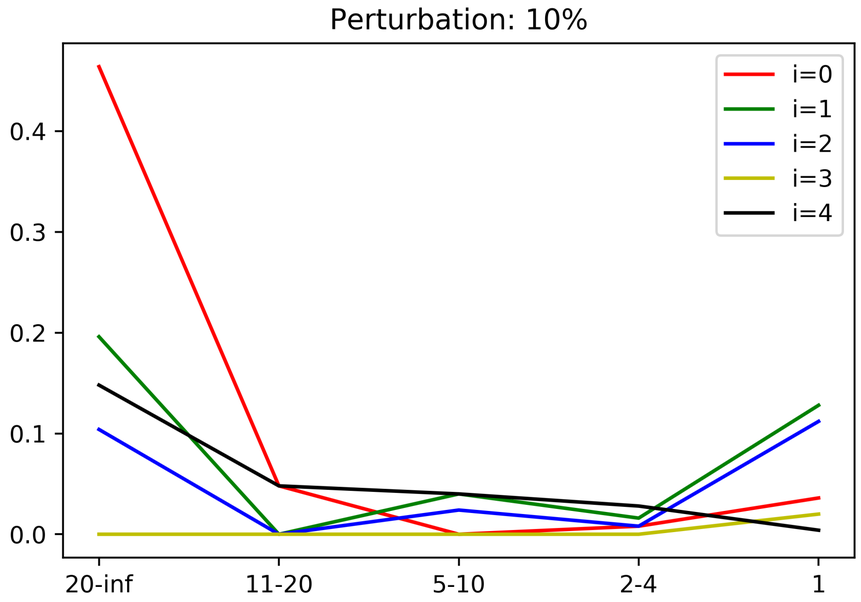
Results by 0% & 2% perturbation





Results by 5% & 10% perturbation

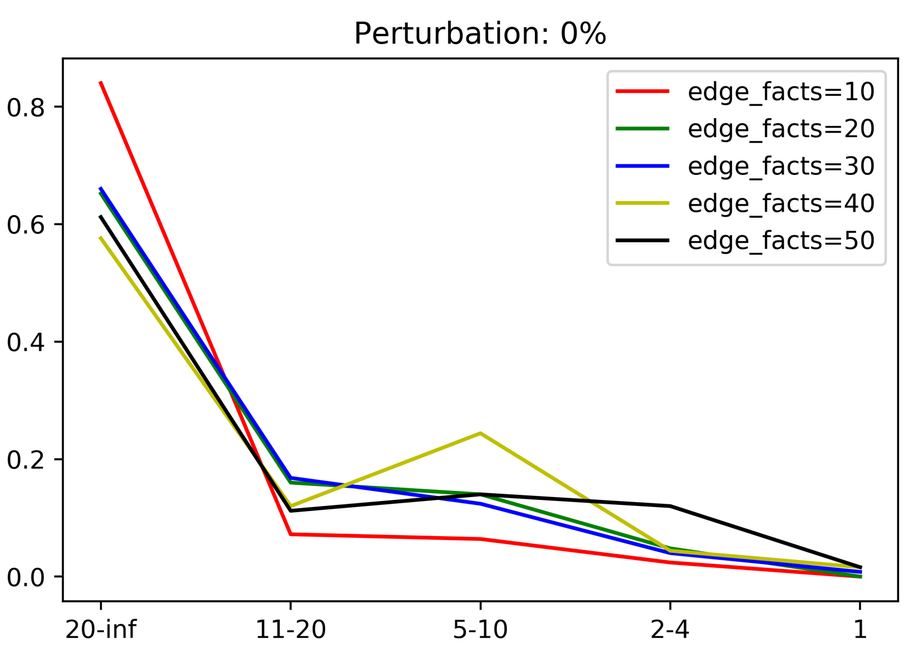


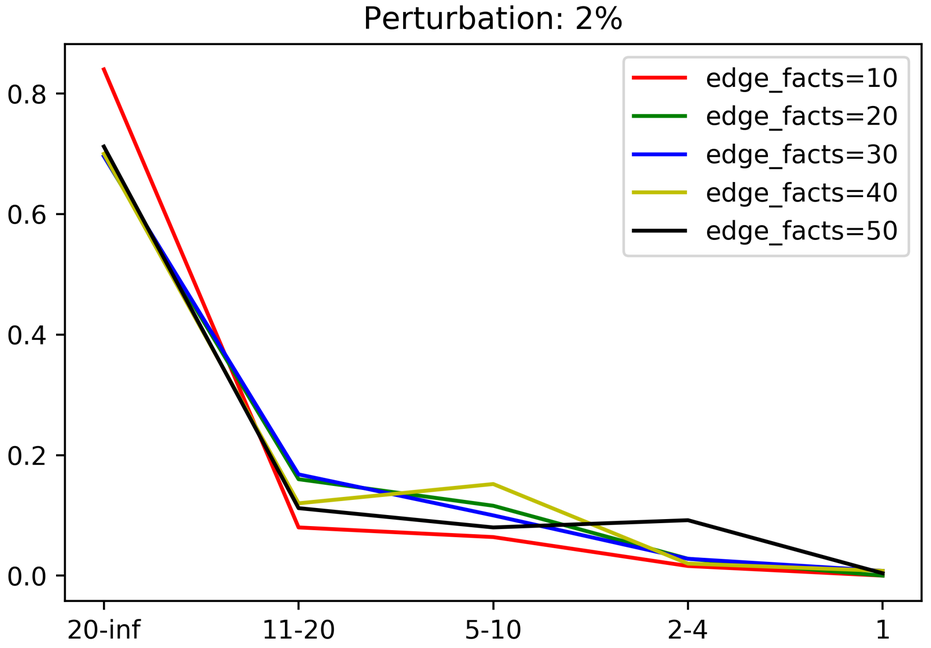


De-anonymization

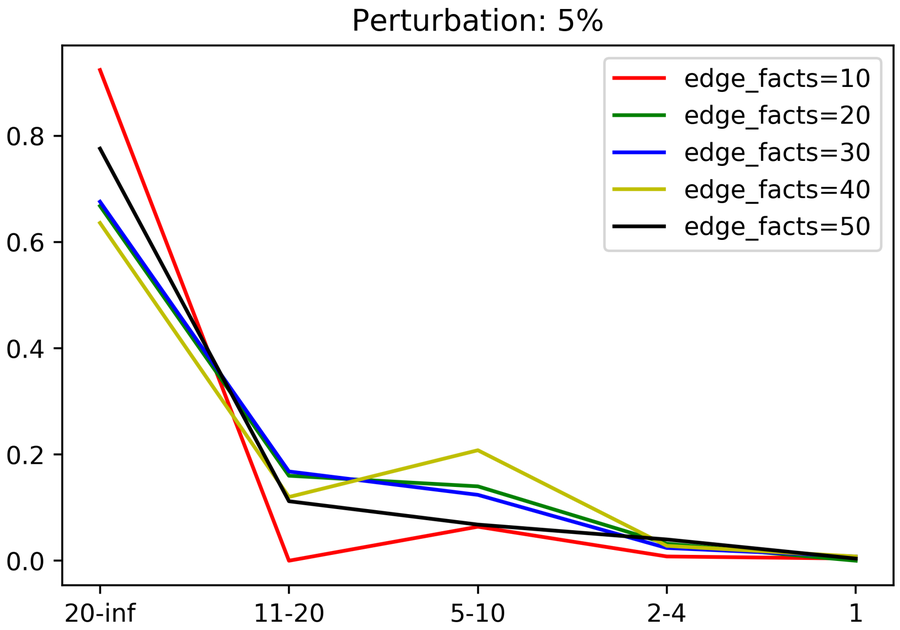
Subgraph technique

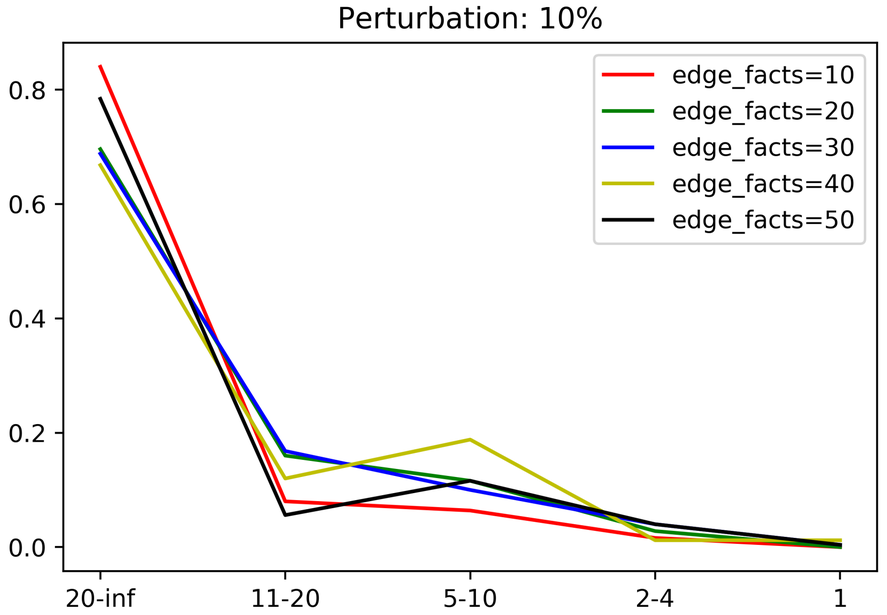
Results by 0% & 2% perturbation





Results by 5% & 10% perturbation





Conclusion

- We have presented some of the main concepts of Anonymization for social network
- We proposed 2 techniques for de-anonymized graph
- Finally we experiment our achievements in python and visualize it in comparison manner

Reference:

Anonymizing Social Networks
Michael Hay, Gerome Miklau, David Jensen,
Philipp Weis, and Siddharth Srivastava
{mhay,miklau,jensen,pweis,siddharth}@cs.umass.edu
University of Massachusetts Amherst
Computer Science Department
Technical Report No. 07-19

Thank you