k-Anonymity for Social Networks and Graphs

Klara Stokes
(joint work with Vicenç Torra)

Abstract

Data Privacy

- *Scenario:* a **database** needs to be released to third parties for its analysis. The database contains sensitive information about individuals.
- *Solution:* the data is modified (**anonymized** or **masked**) to avoid disclosure of the sensitive information.

A popular approach for protecting data in table form is to mask the data set, so that it satisfies k-**anonymity** – ensuring a certain level of privacy.

Relations between individuals or objects can be represented using **graphs**.

Databases that contain relations between individuals are common in research areas like epidemiology and sociology.

This talk is about k-**anonymity in graphs.**

Table of Contents

Table of Contents

Tables and k-Anonymity

- A database **table** is a collection of records that correspond to individuals or entities.
- A **record** is divided into attributes (name, personal number, weight, etc). Traditionally, attributes are either **public** or **confidential.**
- An attribute with a unique entry for every record is an **identifier**.

Naive anonymization of tables consists in removing identifiers.

Example

Original table:

ID	Age	Place of birth	Place of death	HIV
X463J	45	Casablanca	Barcelona	NO
4473H	45	Copenhagen	Barcelona	YES
5839G	45	Barcelona	Barcelona	YES
3435U	55	Barcelona	Barcelona	NO

Naively anonymized table:

Age	Place of birth	Place of death	HIV
45	Casablanca	Barcelona	NO
45	Copenhagen	Barcelona	YES
45	Barcelona	Barcelona	YES
55	Barcelona	Barcelona	NO

Tables and k-Anonymity

Quasi-identifier

A collection of (public) attributes that is enough for identifying at least one individual in a population is called a **quasi-identifier**. This term was coined by the Swedish statistician Tore Dalenius in 1986.

k-anonymity

A table is k-**anonymous** if every combination of entries in a quasi-identifier is repeated at least k times.

Example (2-anonymity)

Age	Place of birth	Place of death	HIV
45	Copenhagen	Barcelona	NO
45	Copenhagen	Barcelona	YES
55	Barcelona	Barcelona	YES
55	Barcelona	Barcelona	NO

As a result a record can not be linked to a set of less than k individuals, so there is no identity disclosure!

Social Network Data and Graphs

Graphs are frequently used to represent networks.

Social network data, or data containing relations between people, can be represented using a labeled graph: network data with additional data attached.

It is known that the graph structure can be used as a quasi-identifier for this type of data, so anonymous release is complicated.

Example: database based on a survey on sexual behaviour and sexually transmissable diseases.

What is k-anonymity for graphs?

k-Anonymity for Graphs

k-Anonymity is based on the concept of a partition of the records in anonymity classes. Therefore, k-anonymity for graphs should be something like:

Sketch of how to achieve k-anonymity for graphs

Classify vertices according to property P. Replace the vertices with an aggregate value (e.g. a median).

Actually, it was observed by Lorrain and White already in 1971 that the computationally correct quasi-identifier (i.e. P) for social networks is the neighborhood of the vertices.

However, this result was never discussed in the context of data privacy and the concept of quasi-identifier was not yet defined then.

k-Anonymity for Graphs

Several suggestions in the literature for the correct choice of property P.

- Vertex degree [Liu and Terzi 2008];
- Local neighborhood structure around vertex [Zhou and Pei 2008];
- Distance to a set of vertices with high degree and betweenness centrality (hubs);
- Graphs metrics or structural properties in general.

There are also approaches in which the edges are clustered instead of the vertices.

Important observation: a graph that is k-anonymous with respect to one quasi-identifier P may fail to be so for another one.

Table of Contents

Graphs

First things first: **what is a graph?**

Graph

A graph is a set of **vertices** and a set of **edges** connecting pairs of vertices.

A graph is **simple** if it has no loops nor multiple edges. Equivalently:

Simple graph

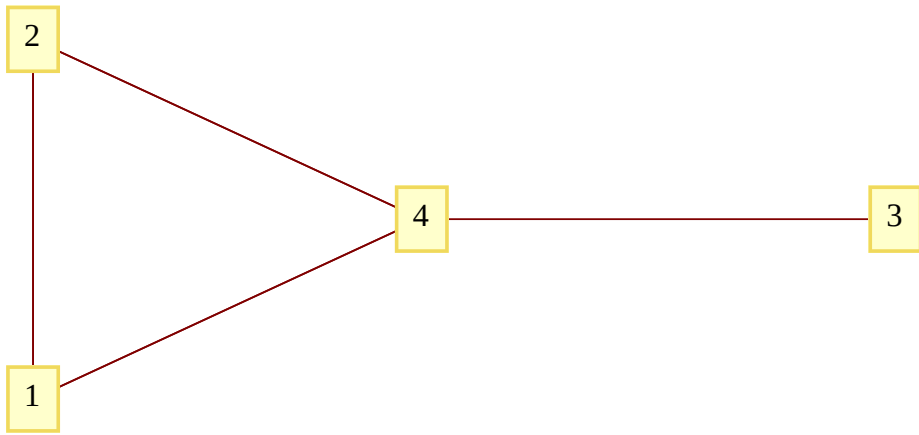
A (simple) graph is a square symmetric **matrix** with entries in $\{0, 1\}$ and 0-diagonal.

This matrix is called the **adjacency matrix** of the graph and is a **lossless representation** of the graph.

If the graph has multiple edges, then the entries in the matrix lives in $\mathbb{N} \cup \{0\}$. The diagonal is zero if and only if there are no loops. A loop is a non-zero entry on the diagonal.

Graphs: A Small Example

	1	**2**	**3**	**4**
1	0	1	0	1
2	1	0	0	1
3	0	0	0	1
4	1	1	1	0



k-Anonymity for Graphs

k-Anonymity for graphs (in terms of records)

A graph is k-anonymous if **every** row (**record**) in the adjacency matrix is **repeated at least** k **times**.

Observe that the matrix is symmetric, so we could have taken the columns instead of the rows.

Every row in the adjacency matrix represents the neighborhood $N(v)$ of a vertex v.

k-Anonymity for graphs (in terms of neighborhoods)

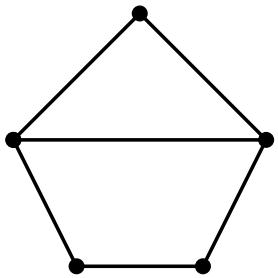
A graph is k-anonymous if every vertex has the **same neighborhood** as at least $k-1$ other vertices.

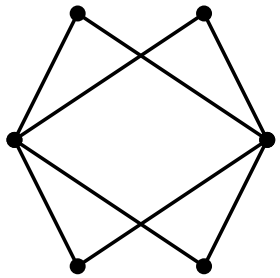
Open and Closed Neighborhoods

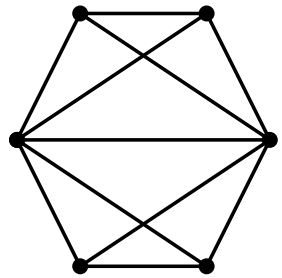
- The open neighborhood of a vertex $v \in V$ is the set $N(v) = \{u \in V : (v, u) \in E\}$.
- The closed neighborhood of v is $\overline{N(v)} = N(v) \cup \{v\}$.
- Two neighborhoods are automorphically equivalent if there is a permutation of the vertices, preserving adjacencies, such that one neighborhood is sent to the other.

Graphs that are k-anonymous with respect to these quasi-identifiers are different.

Examples







The graph on the left is not k-anonymous with respect to open neighborhoods, closed neighborhoods, nor automorphisms, for any $k > 1$.

The graph in the middle is 2-anonymous with respect to open neighborhoods, but not k-anonymous with respect to closed neighborhoods for $k > 1$. It is also 2-anonymous with respect to automorphisms.

The graph on the right is 2-anonymous with respect to closed neighborhoods and automorphisms, but not k-anonymous with respect to open neighborhoods for $k > 1$.

Closed vs Open Neighborhoods

What is the correct choice between these quasi-identifiers?
Depends on the type of application!

Reflexive relation in a network is represented by a self-loop at each vertex.

Reflexive relation: graph with loops $\Rightarrow$ QI: closed neighborhoods.

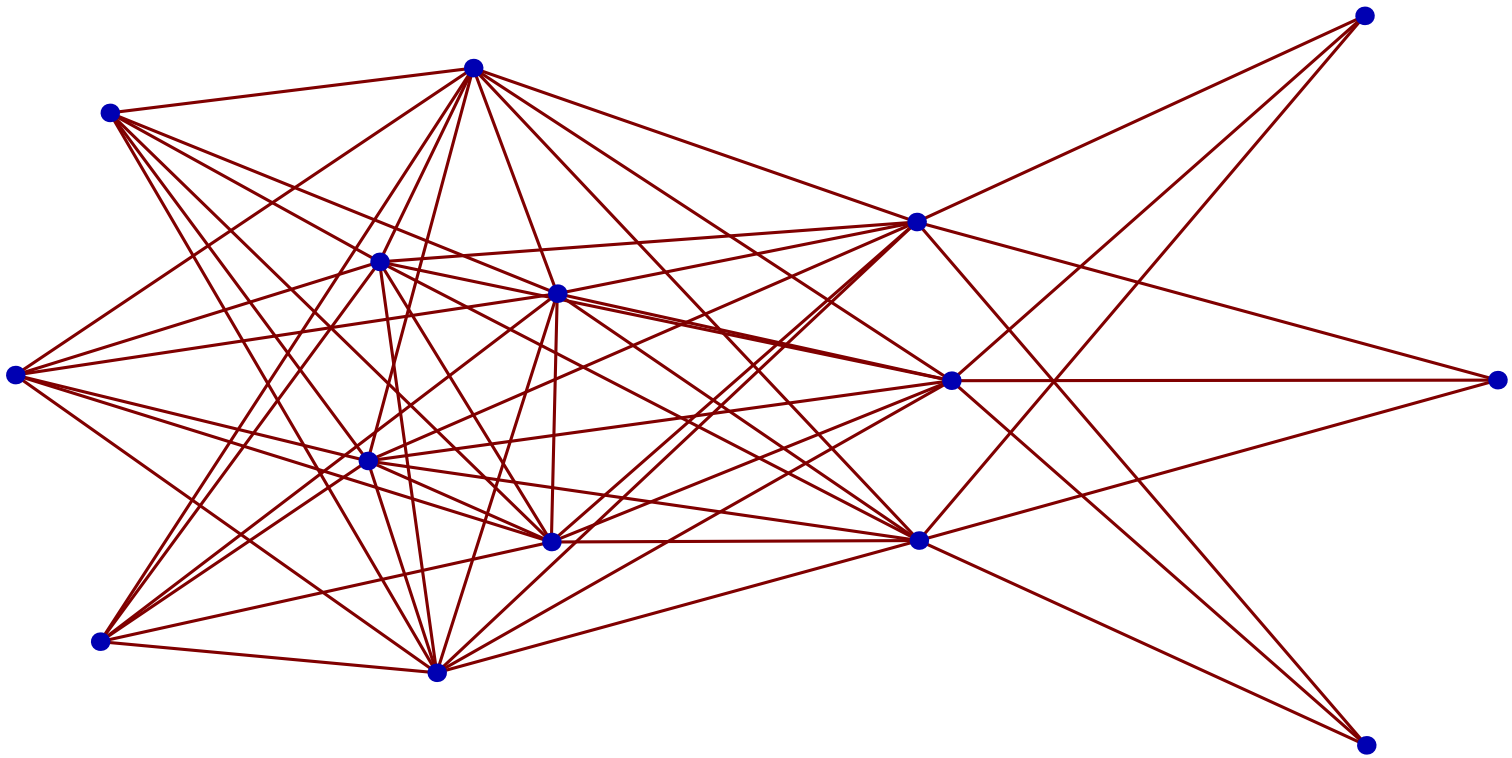
Non-reflexive relation: graph without loops $\Rightarrow$ QI: open neighborhoods.

The "social graph" is typically defined as a graph without self-loops.

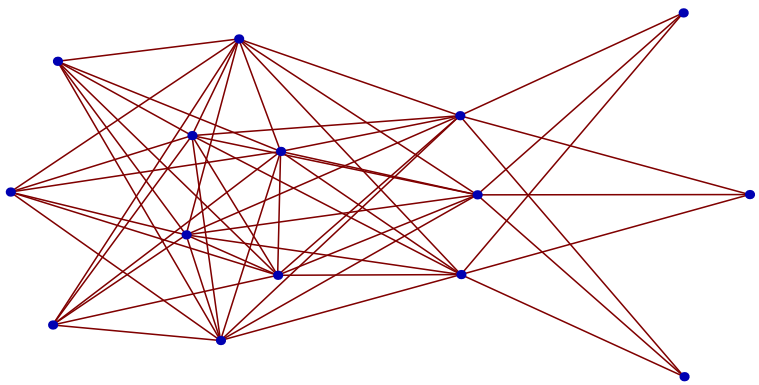
Two vertices u and v in G are **structurally equivalent** if u relates to each vertex in exactly the same way as v does. Then u and v are absolutely equivalent/substitutable within the graph. Open/closed neighborhoods is the strictest QI.

Two vertices with the same neighborhood share the same degree, centrality, etc.

Example: A 3-Anonymous Graph (Open Neighborhoods)



Example: A 3-Anonymous Graph (Open Neighborhoods)



0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0

Example: A 3-Anonymous Graph (Open Neighborhoods)

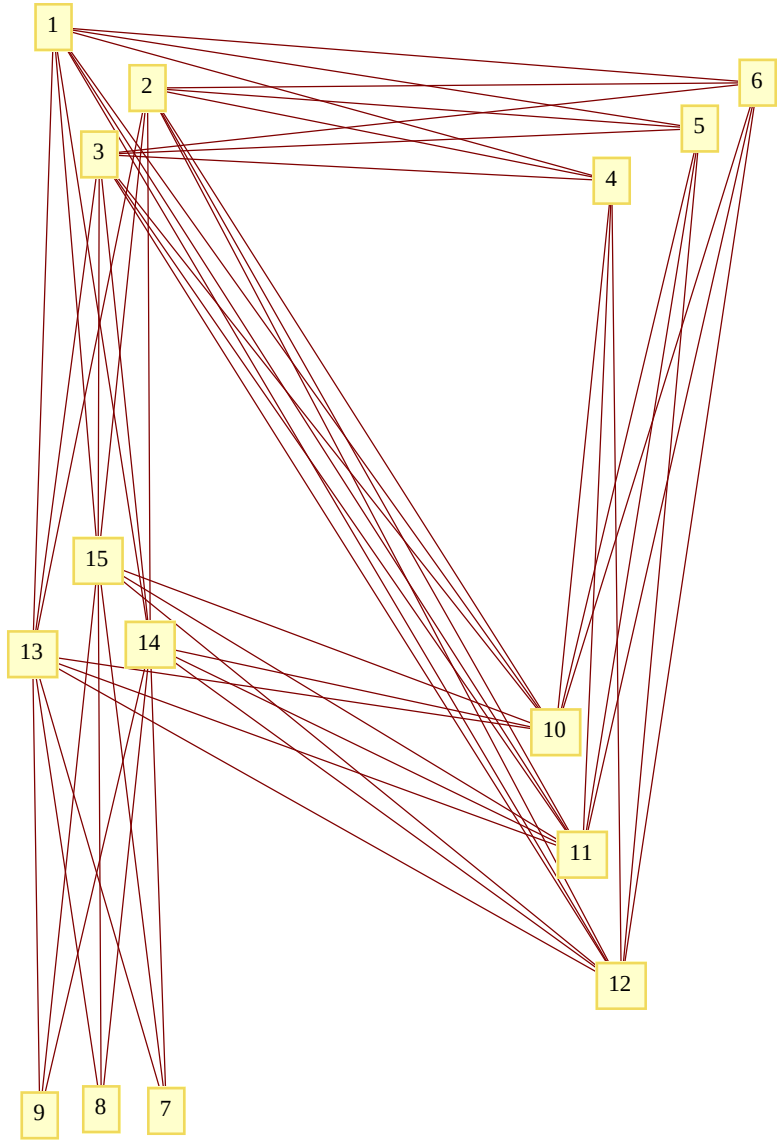
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0

Example: A 3-Anonymous Graph (Open Neighborhoods)

0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1	0	0	0

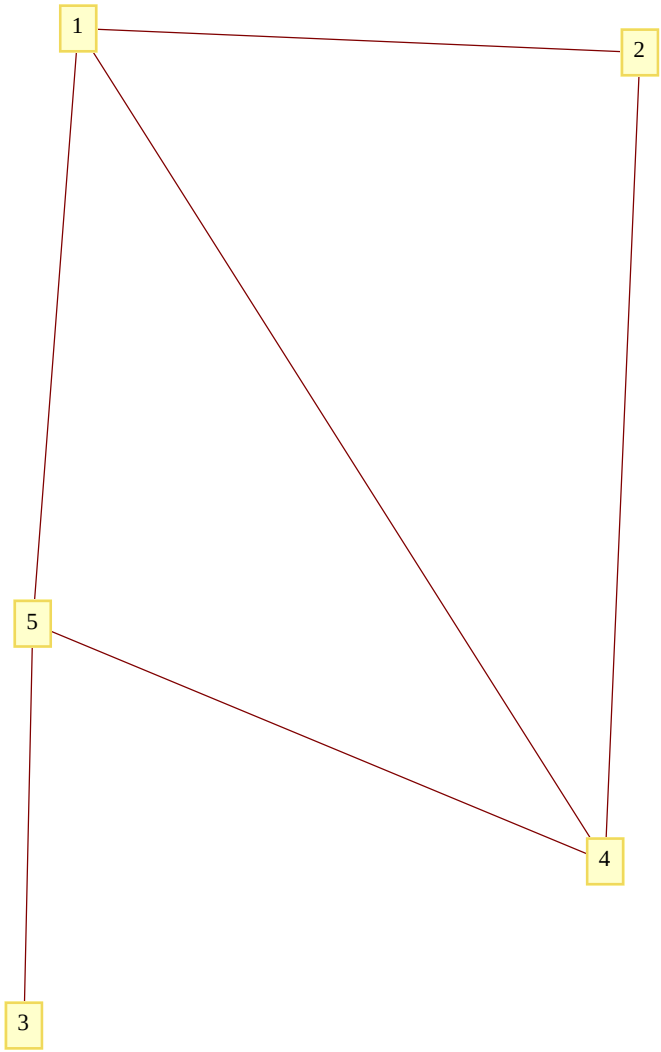
Example: A 3-Anonymous Graph (Open Neighborhoods)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
2	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
3	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
4	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
5	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
6	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
9	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
10	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
11	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
12	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
13	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
14	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
15	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0



Example: A 3-Anonymous Graph (Open Neighborhoods)

	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
2	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
2	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
2	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
4	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
4	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
4	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
5	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
5	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
5	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0



k-Anonymous Graphs: Properties

- Vertices in the same cluster are NOT connected;
- A vertex v in the cluster A is connected to a vertex u in cluster B if and only if ALL vertices in A are connected to ALL vertices in B;
- The number of non-isomorphic k-anonymous graphs on n vertices equals the number of non-isomorphic graphs on c vertices, where c is the desired number of clusters.
  (If all clusters have k vertices, then $c = n/k$.)

k-Anonymous Graphs: Properties

The degree $d(v)$ of a vertex v is the number of neighbors of v: $d = \sharp N(v)$.

Proposition

Let G be a k-anonymous graph. Then:

- We have $k \leq d$ where $d = \min\{d(v) : v \text{ vertex of } G\}$ is the minimum degree;
- There is a partition of the vertex set $V = V_1 \cup \cdots \cup V_m$ such that:
  - For all i, $|V_i| \geq k$;
  - For all $u, v \in V_i$ we have $N(u) = N(v)$;
  - For all $u \in V_i$ and $w \in V_j$, $i \neq j$, we have $N(u) \neq N(w)$. However, in general $N(u) \cap N(w) \neq \emptyset$;
- For any v there is a partition of $N(v) = V_x \cup \cdots \cup V_y$, so that these classes are classes from the partition $V = V_1 \cup \cdots \cup V_m$.

Table of Contents

Achieving k-Anonymity in Graphs

Given any graph G we want to construct a k-anonymous graph G_k based on G.

We provide an algorithm for this purpose.

k-Anonymization of graph

1. **Cluster** the vertices with respect to neighborhoods.
2. **Disconnect** all vertices within the same cluster.
3. For each pair of clusters C_1, C_2:
   - If more than $t\%$ of pairs of vertices from C_1 and C_2 are connected, then **connect** all $v \in C_1$ with all $u \in C_2$.
   - Else **disconnect** all $v \in C_1$ from all $u \in C_2$.

Information loss will depend on the quality of the clustering technique applied.

Modifications to the Algorithm: Closed Neighborhoods

The algorithm can be modified to produce a graph that is k-anonymous with respect to **closed neighborhoods.**

- Ensure you use a clustering algorithm adapted for closed neighborhoods.
- Connect all vertices within the same cluster, instead of disconnecting them.

Modification to the Algorithm: Preserving Existing Edges

The algorithm can be modified to produce a k-anonymous graph **without removing any existing edges.**

- *Application:* User-privacy for friendships in social networks.
- *Problem:* If clustering is not good enough then the anonymized graph is complete (i.e. everyone connected with everyone)!

Clustering in Graphs

Data set: the row vectors in the adjacency matrix of the graph.

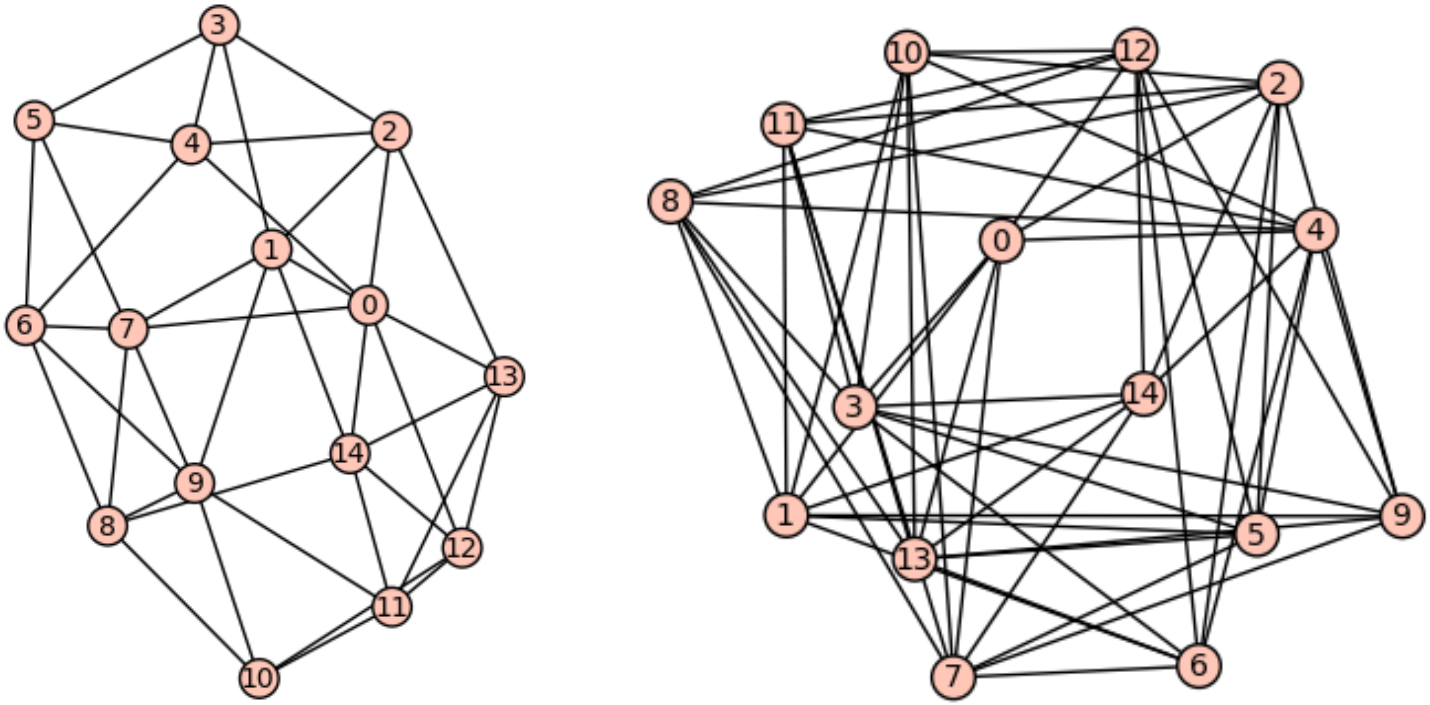
Distances between u and v for neighborhoods give similarities for clustering:

- The **Manhattan similarity** $\sum 1 - \text{xor}(u_i, v_i) = \sum 1 - |u_i - v_i|$. Measures differences both in neighbors and non-neighbors.
- The **2-path distance** $\sum u_i v_i$. Measures only differences in neighbors.

Clustering algorithms for open neighborhood clustering:

- **k-Means** algorithm can be modified for distributed execution. I developed a distributed message-passing version of the k-means algorithm for applications to user-privacy.
- **Mountain clustering:** Outperforms k-means, in particular if we want to preserve existing edges.

Example



The 7-anonymous graph on the right was obtained through k-anonymization of the graph on the left, using our algorithm for k-anonymization of graphs and distributed k-means.

Example

$$\begin{pmatrix}
011010010000111 \\
101100010100001 \\
110110000000010 \\
011011000000000 \\
101101100000000 \\
000110110000000 \\
000011011100000 \\
110001101100000 \\
000000110110001 \\
010000111011000 \\
000000001101100 \\
000000000110111 \\
100000000011011 \\
101000000001101 \\
110000001001110
\end{pmatrix}
\qquad
\begin{pmatrix}
011110010000110 \\
100001101111001 \\
100001101111001 \\
100001101111001 \\
100001101111001 \\
011110010000110 \\
011110010000110 \\
100001101111001 \\
011110010000110 \\
011110010000110 \\
011110010000110 \\
011110010000110 \\
100001101111001 \\
100001101111001 \\
011110010000110
\end{pmatrix}$$

The adjacency matrices of the two graphs from the previous slide.

Preliminary Experiments (Small Graphs and k=3)

$\|V\|$	$SSDE/\|V\|$	$ED/\|V\|$
30	4.13	4.16
60	5.56	6.68
120	6.64	10.75

Barabási-Albert graphs

$\|V\|$	$SSDE/\|V\|$	$ED/\|V\|$
30	3.93	3.68
60	4.51	4.95
120	4.61	6.6

Newman-Watts-Strogatz graphs

SSDE: Sum of symmetric difference between centroids and neighborhoods (normalized by total number of vertices in graph).

ED: Edge difference – the number of edges added or removed (normalized by total number of vertices in graph).

Information Loss

If the original graph is far from being k-anonymous, then information loss will be high.

This is likely to occur if we demand $k >> 0$ (observe $k <$minimum degree of the protected graph).

In such situations, it is useful to **relax k-anonymity.**

Table of Contents

(k,l)-Anonymity

Assume that an adversary, trying to reidentify a vertex in a protected graph has limited knowledge about the original graph structure.

This scenario allows for a weaker definition of k-anonymity.

(k,l)-Anonymity

Let $N(v)$ denote the vector in the adjacency matrix that represents the neighborhood of v.

(k,l)-Anonymity for graphs (I)

A graph on n vertices satisfies (k, l)-anonymity (I) if for any vertex v and for all subset of indices $I \subseteq [1, n]$ of cardinality $|I| \leq l$, there are at least k distinct vertices $\{u_j\}_{j=1}^{k}$ such that $N(u_j)$ equals $N(v)$ over I.

Assume the adversaries knowledge is restricted to a subgraph on l vertices of the original graph, protected in order to satisfy (k, l)-anonymity (I).

Then the adversary cannot reidentify a vertex further than to a set of k vertices.

(k,l)-Anonymity

Now let $N(v)$ denote the set of vertices in the neighborhood of v.

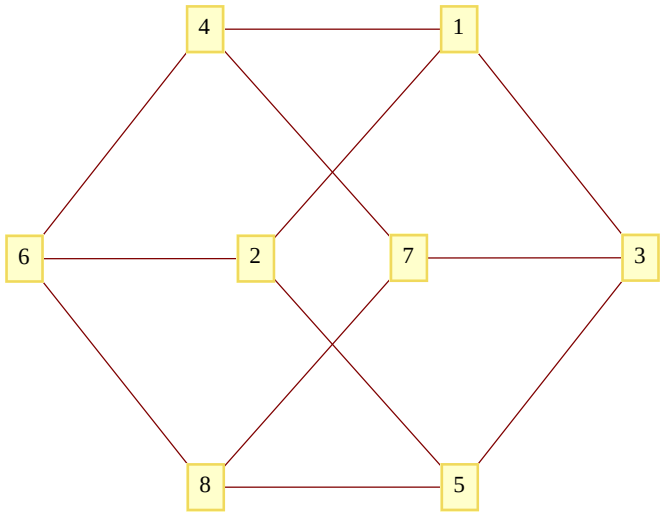
(k,l)-Anonymity for graphs (II)

A graph satisfies (k, l)-anonymity (II) if for any vertex v and for all subset of vertices $U \subseteq N(v)$ of cardinality $|U| \leq l$, there are at least k distinct vertices $\{u_j\}_{j=1}^{k}$ such that $U \subseteq N(u_j)$.

Now assume that the adversaries knowledge is restricted to a subgraph of the original graph with at most l vertices neighbors to any vertex.

If the protected graph satisfies (k, l)-anonymity (II), then the adversary cannot reidentify a vertex further than to a set of k vertices.

Example: (k,l)-anonymity



k-Anonymity

This graph is not k-anonymous for $k > 1$.

Only neighbors (II)

k	l
3	1
2	2
1	3

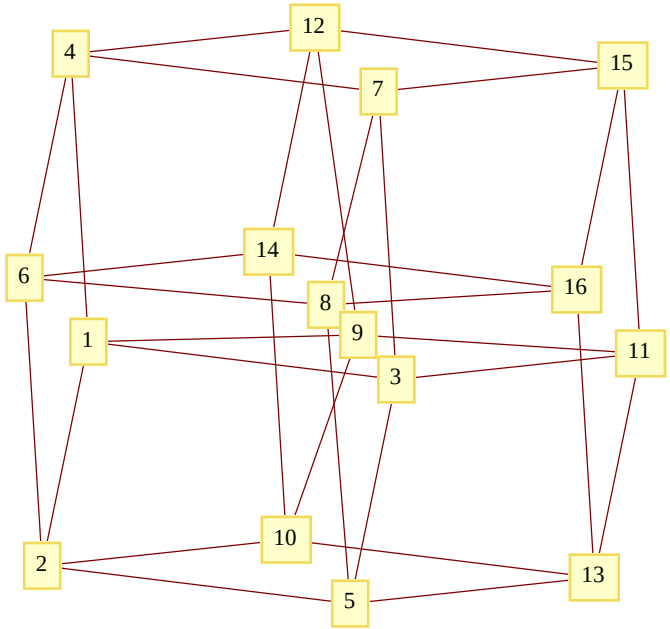
-Anonymous

Both neighbors and non-neighbors (I)

k	l
3	1
2	1
1	8

-Anonymous

Example: (k,l)-anonymity



k-Anonymity

This graph is not k-anonymous for $k > 1$ either.

Only neighbors (II)

k	l
3	1
2	2
1	4

-Anonymous

Both neighbors and non-neighbors (I)

k	l
3	1
2	2
1	16

-Anonymous

Summary

We have

- Proposed a new definition of k-anonymity for graphs;
- Provided a characterization of the k-anonymous graphs in both the regular and the non-regular case;
- Proposed a relaxation of k-anonymity for graphs, called (k, l)-anonymity, that provides equivalent protection as does k-anonymity, when the knowledge of the adversary is limited;
- Provided algorithms for:
  - Transforming a given graph into a k-anonymous graph;
  - Determining the degree of (k, l)-anonymity of a given graph (not described in this talk);
  - Increasing the degree of (k, l)-anonymity of a given graph (not described in this talk).

Conclusion and Future Work

Social network data and other data, containing relations between individuals, can be represented by a labeled graph.

We can anonymize this labeled graph by k-anonymizing the unlabeled graph structure and apply a non-reversible data masking technique to the information in the labels of the vertices in each vertex cluster, independently.

We suggest that the data in each of these clusters could be

- swapped, attribute by attribute, or
- randomly generated using the same distribution as the original data in the cluster,

but we leave the exact determination of suitable methods for future work.

Thank you!

- Stokes, K. and Torra, V. (2012) *Reidentification and k-anonymity: a model for disclosure risk in graphs.* Soft Computing, Springer, 16:10, pp 1657–1670.
- Stokes, K. (2012) *k-Anonimidad para grafos.* Proceedings of XII Spanish Meeting on Cryptology and Information Security (RECSI 2012), Donostia-Sant Sebastián, Euskadi, Spain, 4–7 September 2012.
- Stokes, K. and Torra, V. (2012) *Multiple releases of k-anonymous data sets and k-anonymous relational databases,* International Journal of Uncertainty Fuzzyness and Knowledge-Based Systems, 20:6, pp. 839
- Stokes, K. (2013) *Graph k-Anonymity through k-means and as modular decomposition*. Proceedings of the 18th Nordic Conference on Secure IT Systems (NORDSEC), Ilulissat, Greenland, 18-21 October 2013, Lecture Notes in Computer Science, Springer, in press.