Computational anonymity and some combinatorics

Klara Stokes

Interdisciplinary Workshop on Data Privacy

Maynooth, September 28, 2015

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.

In case it can be assumed that the adversary has certain limitations in memory or in computational power, k-anonymity can be relaxed without affecting the privacy level.

I will show how this is possible and discuss some related combinatorics.

Table of Contents

1. Introduction
2. (k, ℓ)-Anonymity
3. k-Anonymity in graphs

Table of Contents

1 Introduction

2 (k, ℓ)-Anonymity

3 k-Anonymity in graphs

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). In the context of *k*-anonymity, 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.

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 any quasi-identifier is repeated at least k times.

As a result a record can not be linked to a set of less than k individuals (**its anonymity set**) so there is no reidentification.

k-Anonymity provides **unconditional anonymity** if the quasi-identifiers are correctly determined.

But what exactly does correctly determined mean?

Unconditional (theoretical) anonymity requires all public attributes of the table to be considered quasi-identifying in combination with each other.

So, strictly speaking, a k-anonymous table has at least k copies of each record (when restricted to the set of public attributes).

Assumption.
Let T be the table we want to protect. Assume the adversary only has information about at most ℓ of the attributes of each individual in T . (The ℓ attributes do not have to be the same for different individuals.)

Definition. A table T satisfies (k, ℓ)-anonymity if it is k-anonymous with respect to every subset of attributes of cardinality at most ℓ.

Example.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0
2	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0
3	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
4	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
5	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
6	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
7	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
8	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
9	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
10	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
11	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0
12	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0
13	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0
14	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0
15	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0
16	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0
2	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
3	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
4	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1
5	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
6	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0
7	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0
8	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0
9	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0
10	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0
11	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0
12	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1
13	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
14	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0
15	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1
16	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0

Assumption.
Let T be the table we want to protect. Assume the adversary only has information about at most ℓ of the attributes of each individual in T . (The ℓ attributes do not have to be the same for different individuals.)

Proposition. Let T be a table and let $T_{k\ell}$ be a (k, ℓ)-anonymous table that is (somehow) based on T. Under the above Assumption, we have that $T_{k\ell}$ offers the same degree of anonymity as would a k-anonymous table T_k based on T.

The nature of the assumption makes (k, ℓ)-anonymity to a concept of

computational anonymity,

to be contrasted with the

unconditional or **theoretical anonymity**

of k-anonymity.

Goal: an algorithm which, given a table T, outputs a (k, ℓ)-anonymous table, similar to T.

Tool: **Hypergraphs**.

A **hypergraph**, or a **set system**, is

- a set P, with elements called points or vertices, and
- a subset E of the power set of P, with elements called edges

(A hypergraph with two points in each edge is a graph.)

The **degree** of a point p is the number of edges containing p.

The **rank** of an edge e is the number of points in e.

Let

- X be the set of all record entries of the table T, with both values and metadata, and let
- X_ℓ be the set of all subsets of X of cardinality ℓ.

Define a hypergraph $H(T) = (P(T), E(T))$:

- Points $P(T)$: the elements in X_ℓ.
- Edges $E(T)$: the records of T.

If the number of records of T is m, then this hypergraph is uniform of rank $\binom{m}{\ell}$, and the degree of each point equals the number of records with the corresponding ℓ entries.

If T is (k, ℓ)-anonymous, then the degree of each point is either 0 or greater than k.

Problem: Given a hypergraph $H(T)$ representing a table, transform it into a hypergraph $\tilde{H}$ such that **the degree of each point is either 0 or greater than** k.

Algorithm.

While $\exists p \in P(T)$: $0 < \deg(p) < k$ **do**

Choose $q \in P(T)$, minimizing $d(N(p), N(q))$;

Generalize values and metadata (globally) for the points p and q, making them one point $p \vee q$;

The **neighborhood** $N(p)$ of a point p is the multiset containing the points on edges with p.

There are several ways to define a distance between two neighborhoods. Example: use cardinality of the symmetric difference of the two sets.

Example.

Assume $k = 2$, $\ell = 3$. Fix

$$p = \{[\textit{married}, \text{ yes}], [\textit{hair colour}, \text{ brown}], [\textit{height}, \text{ 180 cm}]\}.$$

Some neighbours to p:

$$\begin{array}{l} \{[\textit{married}, \text{ yes}], [\textit{height}, \text{ 180 cm}], [\textit{age}, \text{ 34}]\} \\ \{[\textit{married}, \text{ yes}], [\textit{sports}, \text{ taekwando}], [\textit{age}, \text{ 34}]\} \\ \{[\textit{myopia}, \text{ no}], [\textit{sports}, \text{ taekwando}], [\textit{age}, \text{ 34}]\} \end{array}$$

So there is a record [*married*, yes], [*height*, 180 cm], [*age*, 34], [*hair colour*, brown], [*sports*, taekwando], [*myopia*,no].

Say q is a point with very similar neighbourhood:

$$q = \{[\textit{married}, \text{ yes}], [\textit{hair colour}, \text{ blond}], [\textit{height}, \text{ 180 cm}]\}.$$

Generalization (for example):

$$p \vee q = \{[\textit{married}, \text{ yes}], [\textit{hair colour}, \text{ brown or blond}], [\textit{height}, \text{ 180 cm}]\}.$$

Table of Contents

1 Introduction

2 (k, ℓ)-Anonymity

3 k-Anonymity in graphs

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.

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.
- Local neighborhood structure around vertex.
- 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.

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. It is **simple** if it has no loops nor multiple edges.

Equivalently:

Graph

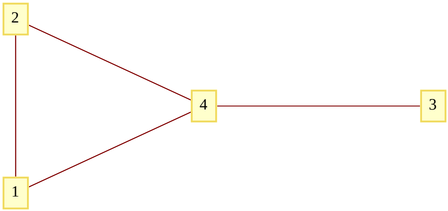
A graph is a square symmetric **matrix** with entries in $\{0, 1\}$. It is **simple** if it has 0-diagonal.

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

- Multiple edges $\Rightarrow$ Matrix entries in $\mathbb{N} \cup \{0\}$.
- Loops $\Rightarrow$ Non-zero entries 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



A simple graph on 4 vertices.

Observe: row v represents neighborhood $N(v)$ of vertex v.

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 $\overline{N(v)} = N(v) \cup \{v\}$.

Example: In a graph representing friendships, my open neighborhood is the set of my friends and my closed neighborhood is the set of my friends and I.

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

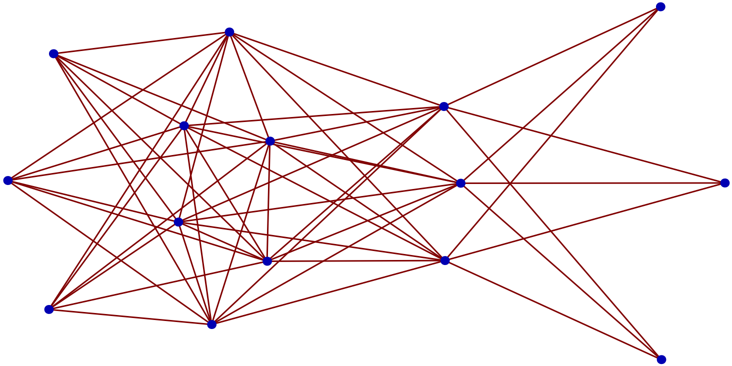
Structural equivalence

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.

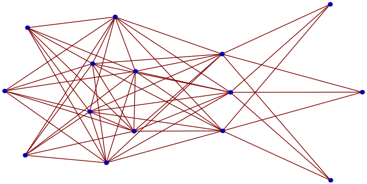
Open/closed neighborhoods is the strictest QI for non-reflexive/reflexive relations.

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)

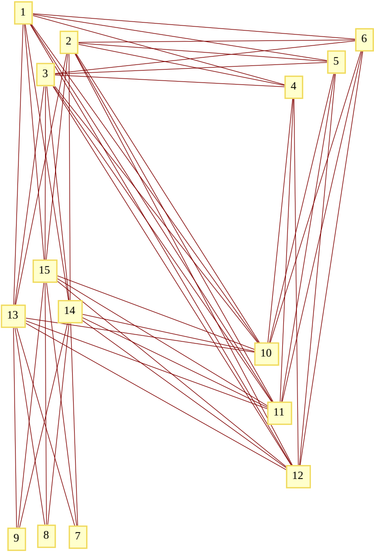
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1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

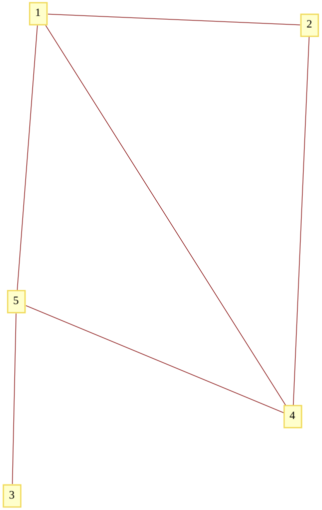
Example: A 3-Anonymous Graph (Open Neighborhoods)

	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



Modular Decomposition of graphs

A **module** in a graph $G = (V, E)$ is a subset of vertices $M \subseteq V$ that share the same neighbors in $V \setminus M$.

A **strong module** is a module that does not overlap other modules.

A congruence partition is a partition of V in which the parts are modules. It is a **maximal modular partition** if the modules are strong and maximal w.r.t. inclusion.

A **factor** is the induced graph on the vertices in one part of a congruence partition.

The modules of a graph define a decomposition scheme for the graph with an associated decomposition tree representing the graphs strong modules.

This tree represents the structure of the graph and is a first step in many algorithms.

k-Anonymous Graphs in Terms of Modular Decomposition

Theorem

Let G be a graph. **If** G **is** k**-anonymous** *with respect to the open (closed) neighborhoods,* **then** G **has a maximal modular partition** $P = \{V_1, \dots, V_m\}$ *such that* $|Vi| \geq k$ *for all* $i = 1, \dots, m$ *and such that the factors of G with respect to P are completely disconnected (complete graphs).*

Efficient way of testing for k**-anonymity in graphs:**

Apply an algorithm for modular decomposition to obtain the maximal modular partition and check that factors are as required.

Relaxing k-Anonymity in Graphs

In general, the factors of the maximal modular partition of a graph can be any graph.

If we do not require factors to be completely disconnected /complete graphs, we get a more flexible definition of k-anonymity, in which only edges between modules are anonymized.

Useful in cases when some edges are more sensitive than others.

Conclusions

We have seen a relaxation of k-anonymity for tables, called (k, ℓ)-anonymity, which is useful when there are many public attributes and it is hard to correctly determine the quasi-identifiers (big data).

We have also discussed k-anonymity in graphs, and related it to the concept of modular decomposition.

Note that (k, ℓ)-anonymity can be applied as it is for graphs. Actually, we first defined it for graphs.

- K. Stokes. Graph k-Anonymity through k-means and as modular decomposition. Proc. of the 18th Nordic Conference on Secure IT Systems (NORDSEC), Ilulissat, Greenland, 18-21 October 2013, LNCS, Springer (2013), pp. 263–278.
- K. Stokes. On computational anonymity. Proc. of Privacy in statistical databases (PSD 2012), Palermo, Italy, 2628 September 2012, LNCS, Springer, (2012), pp. 336–347.
- K. Stokes and V. Torra. Reidentification and k-anonymity: a model for disclosure risk in graphs. Soft Computing, Springer, 16:10, (2012), pp 1657–1670.