
Barriers to the
implementation of
k-anonymity and
related microdata
anonymization techniques
in a realworld application

Barriers to the implementation of k-anonymity and related microdata anonymization techniques in a realworld application

Andreas Wiegand, 1878334
Ludwig Schallner, 1850413

1 Abstract

Bla BLA BLA WIR SIND TOLL!

2 Introduction

Nowadays data is a key factor in nearly every domain. It is comparable to the gold rush of the 19th. century [9]. Furthermore, storage space and network increasingly become affordable [11]. This is leading to the situation that the created and stored data is often not only useful to the original data holder, but to other researchers. Also, some data is only useful if its get shared with other data and get together analyzed. But those data may contain some personal or sensitive information. Such that the data should only get releases if the privacy is protected [7].

Table 1. Basic example

| SSN | Age | Postcode | Problem |
|-------------|-----|----------|-----------------|
| 680-90-2665 | 25 | 4568 | procrastination |
| 008-07-4179 | 34 | 4567 | stress |
| 391-05-7998 | 48 | 4569 | stomach cancer |
| 078-36-3853 | 39 | 4568 | obesity |
| 411-71-9290 | 42 | 4561 | stomach ulcers |
| 527-59-1948 | 27 | 4568 | stress |

Data like table 1 have to get anonymized before release. A very common technique archive that goal is the so-called k-anonymity. Which goal is to prevent the possibility that information about the individual gets leaked. This paper is showing the process of implementing k-anonymity into the real world. In Section 1 we will explain mandatory basic to understand k-anonymity and its purpose. Which leads to Section 2 the theoretical and heuristically implementation of k-anonymity. Section 3 will discuss the underlying barriers of k-anonymity. In Section 4 we will explain to the reader multiple algorithms to implement k-anonymity and its barriers. A summary of the whole paper will be in the last section

3 Basics

Microdata: First of all, it should be clear what microdata is, those data is containing records of information about individuals. The upside versus the more known summary or aggregate data is, that microdata is naturally flexible. Everyone who has this data can perform own statistics from that data [1].

Identifier: Attributes which can identify the record owner explicitly without any other attribute, for example full name (name and surname), telephone number, social security number. nicht sicher ob noch mehr möglich ist [4]

Quasi-identifier: Even though explicit identifier got removed from published data. Attributes which non-explicitly identify the record owner are left. But if they get combined with other non-explicit attributes or other tables, they can reidentify the record owner. In such a case those combination of attributes are called quasi-identifier. For example Gender, Age, Postcode, weight and height [3]. Such process is shown in figure 1.

Sensitive data: Data which is useful for example researchers but are private and should be known publicly nor be accessible for outsiders [8]

Background-knowledge: Because its unknown what the attackers knows, we have to assume additionally to that he have access to table, the attackers knows that the table is generalized (to guarantee k-anonymity). Furthermore the attacks is aware of the domain of the attributes.

Instance-level background knowledge The adversary knows that his target does know specific details about his target. For example Alice (the adversary) knows that Bob do not suffer from some disease, because he does not show the symptoms. In this case the adversary may can conclude what Bob is really suffers from.

Demographic background knowledge Adversary knows for example more general fact for example $P(t[\text{condition}] = \text{cancer} \mid t[\text{Age}] \geq 40)$. The attacker may use it to interference about records [8]

K-Anonymity The goal of making a k-anonymized table, is to have at least (k-1) tuples of each identical tuple taking the corresponding quasi-identifiers into account [11, 7]. For example the 2-anonymized version of the table 1 in the introduction section.

Equivalence class Is a set of all tuples with the identical quasi-identifiers of a table [7].

global recoding/domain generalisation This generalization technique is very common, if a attribute value get generalized then all occurrences of that value gets replaced by the generalized one [11, 10, 7, 6].

local recoding This coding strategies works differently from the above described one. Local recording generalizes attribute values in cells. Because of that this strategies doenst over generalize the table and the data distortion is significantly lower [7].

4 Theoretical and heuristically implantation of K-Anonymity

Another problem we will introduce is, that the producing of k-anonymity of a computational view is an NP-hard problem, like Meyersond and Williams shown.

5 Underlying Barriers

In the following section, we will show the basic and most challenging barriers to the implementation of k-Anonymity. First, we will show the barrier which appears if you k-anonymize the data, the so-called **distortion** of data, in some papers it also mentioned as data loss.

6 Distortion of data as Barrier

A basic underlying barrier of k-anonymity is, how to measure if a implantation has been successful or leads to a satisfying result. This can be measured by a simple calculation. The **modification rate** is representing the fraction of cells which got modified within the attribute set of the quasi-identifier [7].

Table 2. a: original table,b: example for local recording, c: example for domain generalization

| Gender | Birthday | Problem | Gender | Birthday | Problem | Gender | Birthday | Problem |
|--------|------------|---------|--------|------------|---------|--------|----------|---------|
| male | 13.08.1962 | stress | male | 13.08.1962 | stress | * | 196* | stress |
| male | 28.10.1967 | obesity | male | 28.10.1967 | obesity | * | 196* | obesity |
| male | 20.01.1977 | stress | * | 197* | stress | * | 197* | stress |
| female | 15.09.1973 | obesity | * | 197* | obesity | * | 197* | obesity |
| female | 15.03.1985 | stress | female | 15.03.1985 | stress | * | 198* | stress |
| female | 28.05.1986 | obesity | female | 28.05.1986 | obesity | * | 198* | obesity |

Example: for table2 a, the modification rate is $33,33\%$ (4 out of 12 quasi-identifier got changed) for table 2c: its is 100% (12 out of 12 quasi-identifier got changed). Like this simple example shows the modification rate calculation is a

unsatisfying procedure. Because of that the **weighted hierarchical distance** got introduced by Li, Wong, Fu and Pei. To calculate the **weighted hierarchical distance** of a cell, which got generalized from level p to level q , following formula is used [7].

$$WHD(p, q) = \frac{\sum_{j=q+1}^p \omega_{j,j-1}}{\sum_{j=2}^h \omega_{j,j-1}}$$

The WHD for the Age, let the hierarchy of birth date be $\{D/M/Y, M/Y, Y, 10Y, C/T/G/P, *\}$. Where $D/M/Y$ would be day.month.year, $10Y$ a 10 years interval and $C/T/G/R$ for Child/Teen/Grownup/Pensioner.

Example with uniformed weight $w_{j,j-1} = 1$ where $2 \leq j \leq h$ [7]: For the above example Birthday gets generalized from $D/M/Y$ to $10Y$, which corresponds into $WHD_{age}(6, 3) = \frac{3}{5} = 0,6$. For the Gender generalization it would be $WHD_{gender}(2, 1) = \frac{1}{1} = 1$. Which means for generalize 5 cells of age from $D/M/Y$ to $10Y$ one will have the same data distortion as if one generalize 3 cells of Gender from Male/Female to $*$. This calculation shows a much better way to address the distortion of data than the **modification rate** but it does not take how near a generalization is to the root (which would be $*$). BEISPIEL???

Example with height weight: $w_{j,j-1} = 1/(j-1)^\beta$ where $2 \leq j \leq h$ and $\beta = \mathbb{R} \geq 1$ [7]: would be chosen by the user. For example $\beta = 1$. For $WHD_{age}(6, 3) = \frac{0,33+0,25+0,20}{1+0,5+0,33+0,25+0,20} \sim 0,3431$. For $WHD_{gender}(2, 1) = \frac{1}{1} = 1$. The distortion of nearly 3 changed cell of Age from $D/M/Y$ to $10Y$ have the same amout as if one changed one cell of Gender from Female/Male to $*$.

Conclusion Because research need the information out of the tables, like of the examples. Its very important that as less as necessary information gets lost during the anonymization process. To show the importance of this an additionally example, consider a table with survivor of a **idiom disaster beyond all expectations**. Researchers trying to find out the long-time effects of this disaster. Thats why the want to find out if victims get more likely to life a long and happy life if the live far away or close to the disasters location . The algorithm of Ling, Wong, Fu, and Pei which goal is to arrive exactly this will be shown in section [Section einfgen].

6.1 NP Hard

6.2 Attacks as Barrier

Furthermore, also attacks have to be considered as barriers for the implementation, because if the implementation ignores the weaknesses which the attacks use, k -anonymity is useless. It is absolutely necessary that an attacker, under no circumstances, can learn about whatsoever target if he is studying the published database. Not even if the attacker has background knowledge from any other sources [2]. Unfortunately like Dwork showed 2006 that such safety is impossible

because of impossibility of predict what the attacker may know through archiving background knowledge. For example, if the adversary knows that Bob get paid twice as the average German man and the attacker got access to a database which publishes the average income by German men. The anonymity of Bob is compromised even if Bob's data is not in the database [5].

Linking data A barrier to do the implementation of k-anonymity, the attacker can take another dataset and link both together to get rid off the k-anonymity and infer the real individual. This process is called linking data and was first described by Sweeney[11]. She showed that with a example of health care data from 37 states in the USA. The institute from which she bought the data, insures the anonymity of the individuals. Sweeney purchased the voter registration list for Cambridge Massachusettts and received information of the voters including ZIP code, birth date and gender (non explicit identifier) of each voter. She linked that information with the medical data. It was possible to deanonymize the data and get ethnicity, visit date, diagnosis, procedure, medication and total charge of some patients [11].

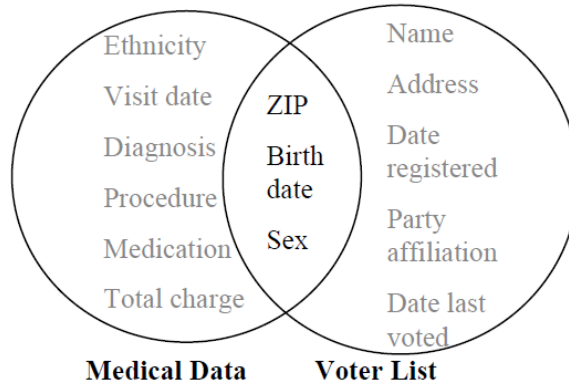


Fig. 1. linking data

You got two datasets A and B. Each dataset got $\langle f_1, \dots, f_n \rangle$ features and $\langle r_1, \dots, r_n \rangle$ rows. Each row is then a tuple r_i with n features $\langle f_1, \dots, f_n \rangle$ describing the individual. Even tho the data is k-anonimized you can get rid of the anonymity of the individual by linking the A to B. So if $A \cap B \neq \emptyset$ it is possible to infer the anonymized individual [11]. As a result any attacker who knows such data (ZIP Code, Birth date and sex) could easily identify with such an attack his victim. For example Peter sees his ex-wife at the doctor, most likely he knows her ZIP-Code, Birth date and sex. Therefore he finds out what she is suffering from.

Unsorted matching attack against k-anonymity There is a possibility of a leak of information, if the release k-anonymity data is in some kind of a sort release. This mean the numerical attributes are descending or ascending sorted and attributes, which be of characters are alphabetical ordered, can give the attacker Information about the sensitive data. To prevent this attack, just get the data into a random order with a pseudo randomized sorting algorithm [11]. As an example take a look at the table 3: matching attack will give an example on that. If you compare the different release generalized tables you can figure out all quasi identifier of those [11].

...

Table 3. matching attack

| Age | ZIP |
|-----|-------|
| 2 | 91058 |
| 4 | 91058 |
| 50 | 27785 |
| 52 | 27785 |
| 20 | 32105 |
| 21 | 32105 |
| 31 | 67676 |
| 32 | 67676 |

| Age | ZIP |
|-----|-------|
| * | 91058 |
| ** | 91058 |
| 5* | 27785 |
| 5* | 27785 |
| 2* | 32105 |
| 2* | 32105 |
| 3* | 67676 |
| 3* | 67676 |

| Age | ZIP |
|-----|-----|
| 2 | 91* |
| 4 | 91* |
| 50 | 27* |
| 52 | 27* |
| 20 | 32* |
| 21 | 32* |
| 31 | 67* |
| 32 | 67* |

Complexity of producing k-anonymity BRauchen wir das???? Till now we only looked at problems of information leaking and privacy problems for individuals. Data is personal-specific information which is structured as a table in rows and columns. Rows a tuple. The columns are attributes with are a set of values which describe the certain attribute. A tuple specify a person. K-anonymity is about protecting the identity of a person not relationships of companies or governments. So the goal of k-anonymity is, not getting more information by linking the data to external data. The bridge between the data and external data is called "quasi-identifier". Examples for that would be ZIP, gender, birth date etc..

Generalization mean, replacing a value with a less specific but semantic identical value. For example we got a list of forenames of buys, (Achmed, Achilles, Achim). To generalize this names you can just (Ach*,Ach*, Ach*) delete the last chars of the name. So there is a less specific domain and now more generalize through this mapping. Suppression on the other hand means not releasing the value at all.

7 Algorithm

7.1 Clustering

Needed because data contains categorical values, the methods are not quite effective. [7].

Algorithm 1: K-Anonymisation by Clustering in Attribute hierarchies (KACA)

```

1 form equivalence classes from the data set
2 while there exists an equivalence class of size < k do
3   | randomly choose an equivalence class  $C$  of size  $k$ 
4   | evaluate the pairwise distance of  $C$  and all other equivalence classes
5   | find the equivalence class  $C'$  with the smallest distance to  $C$ 
6   | generalise the equivalence classes  $C$  and  $C'$ 
7 end
```

7.2 The OLA Algorithm

The OLA Algorithm works in 3 Steps that will now be explained.

1. For every generalization strategy build a binary search to find all k-anonymous nodes in the different strategies. 2. For every generalization strategy that includes k-anonymous nodes save the one with the least Information loss in the hole generalization strategy this is referet to an local optimon k-anonymous solution. 3. Now compare the local optimum solutions to respect of ther information loss. The one with the lowest Information loss of all local optimum solutions is the global optimum solution.

The most time consuming operation is finding the all the K-anonymous notes with and compare them to each other with respect to their information loss. To get a better performance at the Programm step 1) the OLA algorithm works with Predictive Tagging that boost the process. This Tagging take advantage of 2 Theorems of the generalization Lattice. That every k-anonymous note in the same generalization lattice on hight n . All notes above n and in the same generalization strategy are also k-anonymous. So the algorithm only have to find the first k-anonymous note in the strategy and tag all above as k-anonymous.

7.3 Cloaking Algorithm

Moving object data poses new challenges to traditional database, data mining, and privacy-preserving technologies due to its unique characteristics: it is time-dependent, location-dependent, and is generated in large volumes of high-dimensional stream data. The following algorithm shows the a example of privacy production. The Cloacking Alortihm tries to produce Anonymity on location based data for users of Location Bases Services(LBS). The Cloacking Algorithm is installed on a location protection broker on a trusted server and anonymise messages which will afterwards send to the LBS. K-anonymity prevents such a

privacy breach by ensuring that each individual record can only be released if there is at least $k - 1$ other (distinct) individuals whose associated records are indistinguishable from the former in terms of their quasi-identifier values.

there are two possible attacks to get the identity of a sender of a message. At the Restricted Space Identification the attacker A observes that message M is sent from location L afterwards he gets the background knowledge that L belongs to someone specific. For example if Mr. Bob the owner of a flat sends a message and the attacker observes this message. He can re-link the identity of Bob. Another Attack is Observation Identification. If A has observed the current location L of subject S and sends a message M from L then A learns that S has sent M.

To prevent this leaking of information the cloaking algorithm works with Spatial Cloaking and Temporal Cloaking.

Spatial Cloaking's goal is to increase the location of m in such a way that there are more messages in it. So that there is not only one message at a time in an area. Temporal Cloaking extends the sending time until more messages are in one area. A user can give

Spatial cloaking refers to the idea that the coordinates will be boxes and the temporal cloaking refers to the function of delimit the possibility that someone can infer that you are there at that time. With sending the message with a time interval which is bigger than we thought it would be.

8 Summary

Like we saw in Chapter OLA Algorithm. There are possibilities of choose between different Information Loss metrics which all compute different values to the same k -anonymous node. So the implementer has to choose which one fits the most in his data. Different Metrics have pros and cons and have to be compared. Another Problem with Information loss is that you can measure the information loss on your dataset but what would be more interesting is the information loss compared to upcoming data mining tool or machine learning application. The information loss metrics can't know which information is important for the upcoming data mining step. suppression can also harm the quality of the data and should be chosen wisely. The production of k -anonymity is NP-Hard which results in a difficult implementation in real-time applications like the Cloaking - Algorithm had shown. This Complexity can result in problems of finding an optimal solution in real-world data sets and can make a practical implementation difficult. High Dimensional Data like transaction data with more than thousand of attributes are in practice not capable to produce k -anonymity for all attributes. The reason for that is the so called Curse of High Dimensionality which produces a metric space which is too large to produce k -anonymity solutions for all attributes. A Possibility is so called, bounded background knowledge that some attributes can't be used as a quasi-identifier because the effort of getting background knowledge which can be linked to the attributes is estimated too high. Like shown in chapter Attacks as Barrier attacks against k -anonymity are a

threat for the anonymity of the user and can result in identity disclosure. There are solutions regarding these kind of attacks we proposed in this paper. L-Diversity and T-Closeness will help against them. The cloaking algorithm shows a good example of the connection between anonymity and usability. The anonymization of the data reduces the usability of location-based services. Which comes from the construction of the Spatial cloaking and Temporal cloaking boxes to have enough messages to constrain them and let them be k -anonymous. The Software gives the option that a user of the client can decide how much usability he wants to sacrifice to get anonymity. So the user is at least under the control of how much utility he wants to give up.

References

1. Ipumsl-confidentiality, <https://web.archive.org/web/20070823010133/http://international.ipums.org/international/>
2. Dalenius, T.: Towards a methodology for statistical disclosure control. *Statistik Tidskrift* 15, 429–444 (1977)
3. Dalenius, T.: Finding a needle in a haystack or identifying anonymous census records. *Journal of official statistics* 2(3), 329 (1986)
4. Domingo-Ferrer, J., Torra, V.: A critique of k-anonymity and some of its enhancements. In: *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. pp. 990–993. IEEE (2008)
5. Dwork, C.: Differential privacy. In: *Encyclopedia of Cryptography and Security*, pp. 338–340. Springer (2011)
6. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient full-domain k-anonymity. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. pp. 49–60. ACM (2005)
7. Li, J., Wong, R.C.W., Fu, A.W.C., Pei, J.: Achieving k-anonymity by clustering in attribute hierarchical structures. In: *International Conference on Data Warehousing and Knowledge Discovery*. pp. 405–416. Springer (2006)
8. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. pp. 24–24. IEEE (2006)
9. Rossi, B.: Data revolution: the gold rush of the 21st century, <http://www.information-age.com/data-revolution-gold-rush-21st-century-2-123460039/>
10. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05), 571–588 (2002)
11. Sweeney, L.: k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05), 557–570 (2002)