PrioPrivacy: A Local Recoding K-Anonymity Tool for Prioritised Quasi-Identifiers

Alexandros Bampoulidis
Research Studio Data Science, RSA FG
1090 Vienna, Austria
alexandros.bampoulidis@researchstudio.at

Ioannis Markopoulos
Forthnet S.A.
Attika, Greece
jmarkopo@forthnet.gr

Mihai Lupu
Research Studio Data Science, RSA FG
1090 Vienna, Austria
mihai.lupu@researchstudio.at

ABSTRACT

Extensive research in de-anonymisation has shown that in datasets not containing any personally identifying information (PII)—name, address, etc.—individuals can be identified through quasi-identifiers (QIs)—attributes whose combination serves as a unique identifier. In order to deal with this issue, necessary anonymisation measures need to be taken which, however, reduce the quality of a dataset by modifying its values. Data publishers may deem that some QIs are more important than others and, therefore, should be distorted as little as possible in the anonymisation process. Most existing tools do not take such weighting into consideration. In this demo, we present a tool addressing this issue by utilising a local recoding algorithm for k-anonymity, capable of outperforming the existing state-of-the-art tool ARX.

ACM Reference Format:
Alexandros Bampoulidis, Ioannis Markopoulos, and Mihai Lupu. 2019. PrioPrivacy: A Local Recoding K-Anonymity Tool for Prioritised Quasi-Identifiers. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19 Companion), October 14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3358695.3360918

1 INTRODUCTION

Personal data containing information about individuals, but not personally identifying information (PII) such as name, address, etc., is of high importance for research in both academia and industry. Over recent years, privacy concerns have been raised worldwide and strict regulations (e.g. the GDPR[1]) have been introduced addressing explicitly the issue of secure data exchange, even within the same organisation.

Even though personal data does not contain PII, extensive research in de-anonymisation has shown that individuals can still be identified in such datasets and, therefore, it poses a threat to individual privacy. For instance, Sweeney [9] showed that the combination of gender, date of birth, and ZIP code serves as a unique identifier for 87% of the U.S. population. To confront such threats to data publishing, anonymity principles have been introduced. The most prominent of them is k-anonymity [10], which states that every individual in a dataset cannot be distinguished from at least k-1 other individuals - i.e. the maximum probability of de-anonymising any individual is *1 / k*. This is achieved by replacing the original values of a dataset's quasi-identifiers with more general values (e.g. replacing the exact age with an age interval). In general, quasi-identifiers (QIs) are the attributes of a dataset whose combination can serve as a unique identifier for individuals (e.g. gender, date of birth and ZIP code, as in [9]). Several other anonymity principles have been introduced, but in the this paper we focus on k-anonymity. For a broader view on anonymity principles and de-anonymisation we refer to Ji et. al's survey [4].

In order to achieve k-anonymity, generalisation hierarchies need to be defined to describe the general values that could replace the original values of the QIs. These hierarchies consist of at least one level of abstractions (i.e. suppression—the replacement with a generic identifier, typically '*'). Table 1 depicts an example of generalisation hierarchies for 3 attributes. Note that there are other ways to k-anonymise a dataset, such as micro-aggregation [3], but generalisation-suppression is most prominent and therefore the focus of this demonstration.

QIs	Generalisation Levels			
	0 (Original Values)	1	2	3
gender	F or M	*		
Date of Birth	dd/MM/yyyy	MM/yyyy	yyyy	*
ZIP	5 digits	first 4 digits	first 3 digits	*

Table 1: Example of generalisation hierarchies for gender, date of birth and ZIP code. '*' denotes suppression of the original value.

There are two ways (transformation models) of applying generalisation hierarchies: global and local recoding. In global recoding [5], all the values of a QI in the anonymised dataset belong to the same level of generalisation across the whole dataset, while in local recoding [14], different generalisation levels in different subsets of the dataset for each QI may be applied. Naturally, local recoding loses less information, since not all records and their QIs' values in a dataset may need to be transformed in order for them to conform to k-anonymity.

The application of generalisation hierarchies distorts the original values of a dataset and, consequently, decreases the dataset's utility. Algorithms using either global or local recoding try to find a solution that results in as little loss of information (according to some utility metric) as possible and to do so as efficiently as possible. Data utility or data quality metrics measure the information loss incurred from the anonymisation process by measuring the differences between the original and the transformed dataset's QIs'

[1] https://eur-lex.europa.eu/eli/reg/2016/679/oj

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WI '19 Companion, October 14–17, 2019, Thessaloniki, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6988-6/19/10…$15.00
https://doi.org/10.1145/3358695.3360918

values. Various utility metrics are used in the literature, each measuring information loss in a different way, but the most common are iLoss [12] and Non-Uniform Entropy [2, 7].

1.1 Related Work

ARX [8] is the state-of-the-art anonymisation tool, being the most complete in privacy models, data quality metrics and transformation models. To the best of our knowledge, ARX is the only tool that utilises local recoding generalisation and suppression and allows a user to specify priorities of a dataset's QIs, making it the only tool available that we can compare with. It has been stated [8] that it uses the Flash algorithm [5] in finding the optimal solution, according to a given metric, in global recoding, but the authors do not explicitly describe how its local recoding algorithm works. It is actively being developed and its source code is available on GitHub[2].

Other existing anonymisation tools such as UTD[3], CAT [13], Amnesia[4], TIAMAT [1], SECRETA [6], Open Anonymizer[5], have either no support for local generalisation at all, or no consideration of QI priorities. Two tools do support local recoding and QIs' priorities (μ-ARGUS[6] and sdcMicro [11]), but the priorities are only supported for suppression.

1.2 Motivation and Contribution

As just described, most methods and tools currently available to data publishers ignore the fact that some QIs may be more important than others. Such QIs should be distorted as little as possible in the anonymisation process, while still achieving a k-anonymity. The only tool that does local recoding and allows for QI weighting is ARX, but our initial experiments using it found it to be too aggressive in its generalisation/suppression and identified the need for a better algorithm.

Therefore, in this demonstration, we present an application that utilises a local recoding algorithm we defined and implemented that k-anonymises a given dataset and which is capable of outperforming the state-of-the-art anonymisation tool (ARX [8]) when the QIs' importances are specified in the utility metric Non-Uniform Entropy [2] and, specifically, its variant used in local recoding [7].

The rest of the paper is structured as follows: In Sect. 2, we describe the k-anonymisation algorithm we implemented, followed by the experiments' setup and their results in Sect. 3. In Sect. 4, we describe the technical implementation and user interface of our tool and, finally, we conclude with our contribution in Sect. 5.

2 ALGORITHM

We propose a greedy local recoding algorithm that is general in the sense that it does not consider any utility metric as a heuristic, but merely the size of the anonymous groups of records that are formed by applying generalisation rules. The intuition behind it is as follows: Starting from the lowest priority QIs, making as few changes per row as possible, apply first generalisation and then suppression of QIs. The main body of our algorithm is presented in Listing 1, and in this section, we describe the algorithm in detail. Due to space limitation, some of the procedures are only described in natural language in the following paragraphs. However, the source code of our algorithm and the experiments described in Sect. 3 can be found here[7].

The algorithm requires four inputs: the dataset, the prioritised list of QIs, the desired k-anonymity, and the generalisation hierarchies. The prioritised list of QIs contains the QIs accompanied by an integer denoting its priority—the lower the integer, the higher the priority. Specifying QIs having the same priority number is allowed. The generalisation hierarchies consist of rules for each QI in different levels, with each rule being in the form of *{original values} => transformed value*.

In line 2, we identify *unsafeDataset*—the data subset with the records that do not conform to the desired k-anonymity. Then, we apply generalisation and suppression in i iterations, where i is the number of QIs in the dataset. In each iteration we add a QI in the *QIs* list. Note on line 5: If there are QIs with the same priority, then we simply pop the first element of the respective data structure.

In the generalisation procedure, we parse the generalisation hierarchies of the QIs level by level, generate all the possible combinations of rules (line 13) and form anonymous groups with the *unsafeDataset*'s records with as few changes per record as possible. The *genCombinations* variable contains all the possible combinations of rules, ranging from 1- to *QIs.size*-transformation rules. Each combination contains at most 1 rule per QI. An example of a 2-transformation rule is: *[workclass:{local-gov,federal-gov} = > Government],[country:{Austria,Greece} = > Europe]*. Then, starting from the 1-rule transformations, and for each of them, we apply the transformation to a copy of the *unsafeDataset*, calculate the maximum anonymous group size (line 20), and apply the transformation with the maximum anonymous group size (*maxK*) to the dataset, and only to the records that form *maxK*-sized groups. This is done until no anonymous groups can be formed with the 1-transformation rules and, then, do the same for the 2-, 3-, etc. transformation rules.

The suppression procedure has the same rationale as generalisation: suppress as few QIs per record as possible. However, the suppression procedure differs slightly from generalisation: For each row in the *unsafeDataset* and for each suppression rule (*supp*), we check whether the transformed record (*suppRow*) is k-anonymous in the original dataset (lines 34-36) and in line 38, we apply the first suppression rule that is parsed and makes the record k-anonymous, and only to the record that is addressed in line 32. The intuition of the suppression subprocedure can be explained with a simple example: Let us consider a dataset {gender,age,country} with 2 {f,20,AT} records, 2 {m,20,GR} records and 1 {m,20,AT} record. The first four records are 2-anonymous, but the last one is not. In order for it to be 2-anonymous the record must be transformed to {*,20,*}, so that the values of the suppressed QIs could be either {m,AT} or {f,GR}. If, however, the record was to be transformed to {*,20,AT}, then it could be inferred that the person is a male, since there are two possible values for gender and one of them is missing for the pair {20,AT} and by reasoning, it is inferred as "m". This way, the string of the fifth record ({*,20,*}) is not 2-anonymous in the sense that it is a unique string, but in reality it offers 2-anonymity protection. We inferred, by examining ARX's anonymised outputs, that it considers '*' as an independent nominal value, in contrast to our algorithm

[2] https://github.com/arx-deidentifier/arx
[3] http://www.cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php
[4] https://amnesia.openaire.eu/index.html
[5] https://sourceforge.net/projects/openanonymizer/
[6] http://neon.vb.cbs.nl/casc/mu.htm
[7] https://github.com/alex-bampoulidis/prioprivacy

and, therefore, it would not produce the solution described above (it would see {*,20,*} as not 2-anonymous). Since this subprocedure of the algorithm is computationally intensive, the lines 32-39 are executed in parallel - one *unsafeRow* per available CPU thread.

Algorithm 1 PrioPrivacy Algorithm

```
1:  procedure PrioPrivacy(origDataset, priorities, k, genHier)
2:      dataset ← origDataset
3:      unsafeDataset ← createUnsafeDataset(dataset, k)
4:      QIs ← {}
5:      for p = priorities.size; p ≥ 1; p − − do
6:          for each QI : priorities.get(p) do
7:              QIs.add(QI)
8:              generalisation(dataset,unsafeDataset,QIs,k,genHier)
9:              suppression(origDataset,dataset,unsafeDataset,QIs,k)
10:     return(dataset)
11:
12: procedure generalisation(dataset, unsafeDataset, QIs, k,
    genHier)
13:     for level = 1; level ≤ max(genHier(QIs).levels); level++ do
14:         genCombinations ← generatePermutations(genHier,
    QIs, level)
15:         for c = 1; c ≤ QIs.size; c++ do
16:             while TRUE do
17:                 maxK ← −1
18:                 maxKRule ← {}
19:                 for each rule : genCombinations.get(c) do
20:                     temp ← unsafeDataset
21:                     ruleK ← maxKOfTransformation(rule, temp)
22:                     if ruleK > maxK then
23:                         maxK ← ruleK
24:                         maxKRule ← rule
25:                 if maxK < k then
26:                     break
27:                 applyGeneralisationRuleToMaxKRows(dataset,
    unsafeDataset,maxK,maxKRule)
28:
29: procedure suppression(origDataset, dataset, unsafeDataset,
    QIs, k)
30:     suppCombinations ← generatePermutations(QIs)
31:     for c = 1; c ≤ QIs.size; c++ do
32:         for unsafeRow : unsafeDataset do
33:             for each supp : suppCombinations.get(c) do
34:                 suppRow ← applyRule(unsafeRow, supp)
35:                 temp ← origDataset
36:                 isKAnonymous ← checkKAnonymity(suppRow,
    temp, k)
37:                 if isKAnonymous then
38:                     applySuppressionRuleToUnsafeRow(dataset,unsafeDataset,
    unsafeRow,suppRow)
39:                     break
```

3 EXPERIMENTS AND RESULTS

To evaluate our algorithm and compare it with ARX, we used a standard dataset (the UCI Adult dataset) consisting of 30,162 records and 9 QIs. For comparability, the generalisation hierarchies used in the experiments are taken from ARX's GitHub repository.

We conducted four experiments, each one having different QIs priorities. In experiment 1, priority is given to QIs with the least number of distinct values (the QIs {sex, salary-class} and {workclass, marital-status} have the same priority) and experiment 2 has the reversed priorities of experiment 1. In experiment 3, we mix the QIs, alternating most distinct and least distinct QIs (i.e. most distinct, least distinct, 2nd most distinct, 2nd least distinct, etc.) and in experiment 4, alternating least distinct and most distinct QIs (i.e. least distinct, most distinct, 2nd least distinct, 2nd most distinct, etc.). Each experiment was run nine times for k∈[2,10].

Additionally, ARX's settings are as follows: Local transformation using 10^9-1 iterations (maximum), optimising for the Non-Uniform Entropy (NUE) metric and using the arithmetic mean as aggregate function. ARX's QIs' priorities (called weights) are specified as a value between 0 and 1. The integer priorities from our algorithm are converted to ARX weights using the following formula:

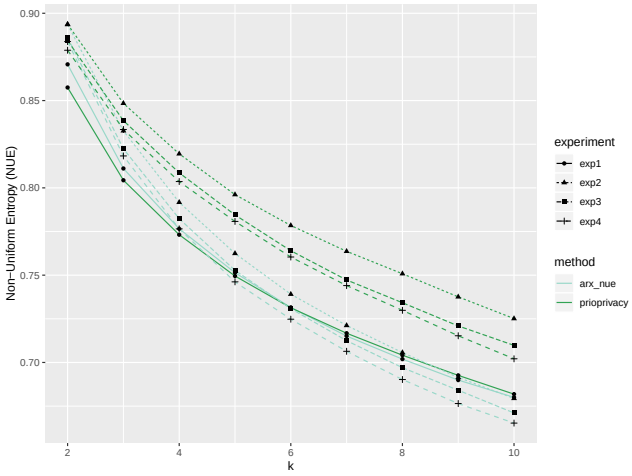

Figure 1: Transformed datasets' quality according to the Non-Uniform Entropy metric [7]

Execution time in minutes				
Method	Experiment			
	1	2	3	4
PrioPrivacy (8 Threads)	1.55	1.97	1.29	1.45
PrioPrivacy (1 Thread)	1.34	4.46	2.23	2.19
ARX NUE	6.18	32.84	20.78	16.61

Table 2: Average execution time per method and experiment (9 runs per experiment)

$$w(priorities, QI) = \frac{\#\{priorities\} - priorities(QI) + 1}{\#\{priorities\}}$$

Figure 1 depicts the quality of the transformed datasets, according to the Non-Uniform Entropy metric [7] (the higher, the better). We outperform ARX in 3 out of 4 experiments and only slightly underperform in experiment 1 until k = 6, and slightly outperform until k = 10. Figure 2 depicts the quality of the transformed datasets per attribute (sorted by their importance) resulting from experiment 1. Despite performing similarly to ARX in this experiment, we still manage to achieve higher quality in top-priority QIs. The attribute-level quality results of the other 3 experiments are similar. Due to space limitations, the respective figures for the other 3 experiments can be found on our GitHub (link provided in Sect. 2). Finally, table 2 depicts the average execution time in minutes of the experiments. On average we achieved a much better execution time in all experiments even when executing our algorithm on one thread.

4 APPLICATION

Our application is a desktop application developed in JavaFX and requires Java version 8+. Figure 3 depicts the user interface (UI) and in this section we describe its elements.

Area 1 contains three elements: **1.** An import button which, when clicked, opens a data import wizard. **2.** A slider indicating the desired k-anonymity of the dataset. **3.** An *Anonymise* button which, when clicked, prompts the user to specify the output directory and file name, and anonymises the dataset accordingly.

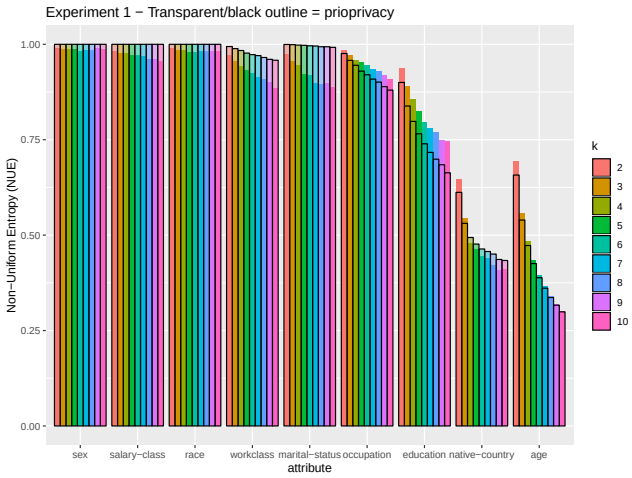

Figure 2: Data quality per attribute according to the Non-Uniform Entropy metric [7] resulting from experiment 1

In area 2, the attributes of the imported dataset are listed. The checkboxes indicate whether the attribute is a QI or not and the numbers indicate the importance of the QIs - the lower the number, the higher the importance. A click on any of the attributes leads to a change in areas 3 and 4.

Area 3 contains two elements: **1.** A list of the unique values of the attribute clicked on area 2. **2.** A text editor where the user can specify the generalisation rules for the selected attribute in the form {value$_1$,...,value$_n$ => transformed value}. The user may import such a file by clicking the *Import* button/tab. Generalisation levels may be added by clicking the + button/tab, and may be removed by clicking the *Remove* button/tab.

Area 4 contains a histogram of all unique values of the attribute clicked on area 2 grouped in *safe* and *at risk*, depending on whether the record they belong to conforms to k-anonymity or not, and helps the user in defining the generalisation hierarchies. The histogram may be resized by dragging the edges and have its coordinates flipped by clicking the *Flip* button.

5 CONCLUSION

In this paper, we presented PrioPrivacy, a k-anonymity tool focused on distorting as little as possible attributes important to the user. It does so by utilising an algorithm we implemented which, despite its simplicity and greediness, can outperform the state-of-the-art ARX anonymisation tool.

ACKNOWLEDGMENT

This research was supported by the Safe-DEED project (H2020, GA No.: 825225), funded by the European Commission.

REFERENCES

[1] C. Dai, G. Ghinita, E. Bertino, J.-W. Byun, and N. Li. 2009. TIAMAT: a tool for interactive analysis of microdata anonymization techniques. *VLDB Endowment* (2009).
[2] T. De Waal and L. Willenborg. 1999. Information loss through global recoding and local suppression. *Netherlands Official Statistics* (1999).
[3] J. Domingo-Ferrer and J. M. Mateo-Sanz. 2002. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering* (2002).
[4] S. Ji, P. Mittal, and R. Beyah. 2017. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials* (2017).
[5] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. 2012. Flash: efficient, stable and optimal k-anonymity. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing.*
[6] G. Poulis, A. Gkoulalas-Divanis, G. Loukides, S. Skiadopoulos, and C. Tryfonopoulos. 2014. SECRETA: A system for evaluating and comparing relational and transaction anonymization algorithms. (2014).
[7] F. Prasser, R. Bild, and K. A. Kuhn. 2016. A Generic Method for Assessing the Quality of De-Identified Health Data. *Studies in health technology and informatics* (2016).
[8] F. Prasser, F. Kohlmayer, R. Lautenschlaeger, and K. A. Kuhn. 2014. Arx - a comprehensive tool for anonymizing biomedical data. In *AMIA 2014.*
[9] L. Sweeney. 2000. Simple demographics often identify people uniquely. (2000).
[10] L. Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* (2002).
[11] M. Templ. 2008. Statistical disclosure control for microdata using the R-package sdcMicro. *Transactions on Data Privacy* (2008).
[12] X. Xiao and Y. Tao. 2006. Personalized privacy preservation. In *SIGMOD 2006.*
[13] X. Xiao, G. Wang, and J. Gehrke. 2009. Interactive anonymization of sensitive data. In *SIGMOD 2009.*
[14] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. 2006. Utility-based anonymization using local recoding. In *SIGKDD 2006.*

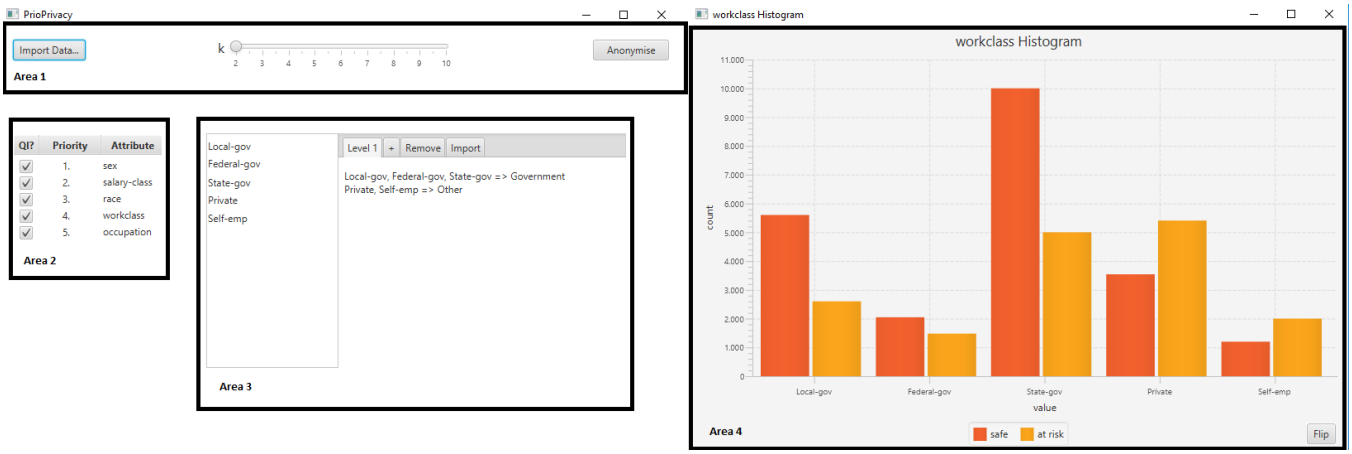

Figure 3: User Interface of PrioPrivacy