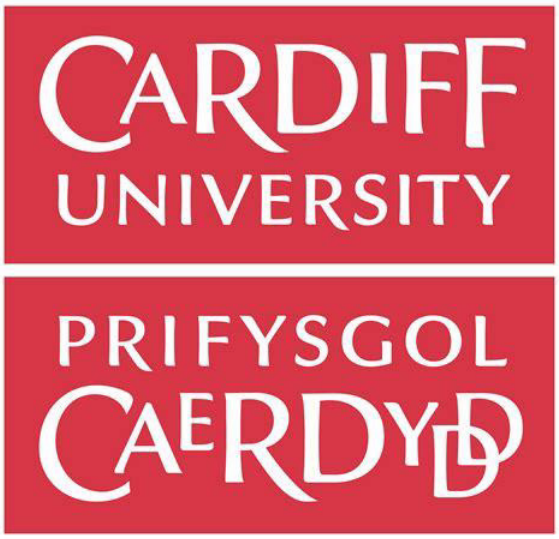

Cardiff University

School of Computer Science and Informatics

Implementation of a Data Privacy Protection Tool for Relational Data

CM3203 Final Year Project – 40 Credits

Initial Plan

Author: Jack Davies

Supervisor: Jianhua Shao

Moderator: Usashi Chatterjee

Academic Year: 2020/21

Description

In the modern world, data is being collected and stored on almost every action and behaviour that people carry out. Everything from your heart rate collected by a wearable smart watch to your purchasing habits online or in the supermarket is stored on a database waiting to be used for various purposes. Due to the rise of smart phones and social media, along with the more traditional personal information that is kept such as medical data and employment history, it would be very challenging to find someone that has avoided having data collected about them in some way.

To protect people's privacy, there exist several laws and regulations such as the European Union's GDPR and the USA's Privacy Act which prohibit the general disclosure of a data record about an individual without permission. This leads us to an issue when it comes to using collected data. In order for this data to be released to third parties, it must be "anonymised" such that no specific record can be linked to an individual.

This project aims primarily to create a tool that will implement the k-anonymity approach (Samarati and Sweeney 1998 [4]) to anonymising a given relational data set. K-anonymity is a property possessed by anonymised data where the information for each person contained in the released data set cannot be distinguished from at least k-1 other individuals in the data set. This can be achieved by ensuring that within a data set, the *quasi-identifier* values are generalised such that each unique value is equal to k-1 other records. Quasi-identifiers are defined as attributes in a table that are available to anyone (e.g., postcode found via voter registration or date of birth via social media profile), which can be used to perform record linkage attacks to discover the identity of the person in the data set. Figure 1 shows two tables illustrating before (a) and after (b) k-anonymisation – in this case k=2. Quasi-identifiers are shown in yellow, sensitive data is shown in blue and an explicit identifier is shown in orange, the explicit identifier is removed from the anonymised table for obvious reasons.

	Name	Age	Postcode	Illness
rec1	Alice	28	CF24 3BW	Flu
rec2	Bob	30	CF24 3BW	Appendicitis
rec3	Connor	29	CF23 5AB	Diabetes
rec4	Denise	29	CF23 7RZ	Flu
rec5	Ethan	30	CF23 7RZ	Broken nose

Fig. 1a

	Age	Postcode	Illness
rec1	[28-30]	CF24 3BW	Flu
rec2	[28-30]	CF24 3BW	Appendicitis
rec3	[29-30]	[CF23 5AB, CF23 7RZ]	Diabetes
rec4	[29-30]	[CF23 5AB, CF23 7RZ]	Flu
rec5	[29-30]	[CF23 5AB, CF23 7RZ]	Broken nose

Fig. 1b

Much research has been done on anonymisation of data (Fung et al. 2010 [2]), but a relatively small amount has been done on the usefulness of the data post-anonymisation. This project will therefore also use a machine learning algorithm to investigate the utility of the anonymised data, as it being useful to third parties is intrinsic in the value of the data.

Expected Approach

In this project, the expected approach taken in implementing the k-anonymity tool will be to use the Mondrian multidimensional algorithm (LeFevre et al. 2006 [3]). The essential idea behind the algorithm is visualised in Figure 2. The input dataset is partitioned based upon the values in the quasi-identifier attribute set such that there exist at least k unique record tuples in each partition. Once a set is partitioned, each partition is then generalised by taking a summary of the partition values for each attribute and creating a summarisation value. The summarisation value for each attribute is then applied to each partition value such that all quasi-identifier tuples in the partition are now identical. Once all partitions have been generalised, they can be merged into a single table which will be a k-anonymisation and can be exported for third party use.

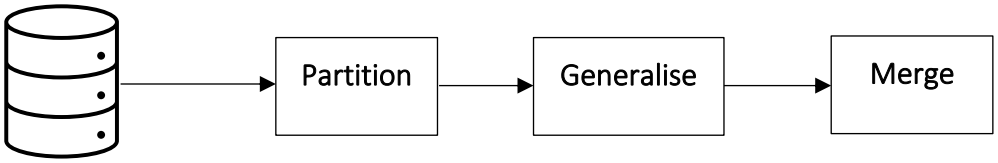

Fig. 2

A given dataset *D* is partitioned in the Mondrian algorithm similarly to a k-d tree (Friedman et al. 1977 [1]). So long as *D* is large enough to be cut into two subsets, each having at least *k* tuples, then a cut is performed along one of the attributes in the quasi-identifier set $\{X_1 - X_n\}$. The attribute selected $X_i \in \{X_1 - X_n\}$ is the attribute with the largest range of values. A frequency set *F* for the values in the domain of X_i is populated and the corresponding values are ordered ascendingly for numerical values, and alphabetically for categorical values. The median *m* of frequency set *F* is then found and the value corresponding to *m* in the ordered set is the pivot value *pv* used for the cut. At this point two methods of partitioning are possible: strict and relaxed. Strict partitioning is where the two subsets of *D* (*lhs* & *rhs*) regarding selected attribute X_i must not have intersecting values. Relaxed partitioning allows intersecting values. The decision on which method to use is left open at this stage. With the strict method, subsets *lhs* and *rhs* are populated with tuples such that *lhs* contains tuples with values ≤ *pv*, and *rhs* contains tuples with values > *pv*, both in terms of the value's position in the ordered list relative to *pv*. Applied recursively to all partitions this will result in a set of multidimensional regions each with at least *k* unique tuples. The pseudocode for the strict algorithm is shown in Fig. 3. The relaxed algorithm would require a minimal change regarding how tuples are added to *lhs* and *rhs*.

Mondrian Algorithm (LeFevre et al. 2006) – Strict Partitioning

```
AnonymiseStrict(dataset D)
if (D cannot be partitioned):
        return D
else:
        X_i = chooseAttribute(D)
        F = frequencySet(D, X_i)
        pv = median(F)
        lhs = {t ∈ D | t.X_i ≤ pv}
        rhs = {t ∈ D | t.X_i > pv}
return AnonymiseStrict(lhs) ∪ AnonymiseStrict(rhs)
```

Fig. 3

The tuples in each region can be generalised using a summarisation. For numerical values, a simple range is sufficient, and it has been proposed (LeFevre et al. 2006. pp. 6-7 [3]) that the mean could also be included, as both range and mean are useful for different purposes. For non-numerical values, there are two options. The first option is to create a taxonomy whereby words are generalised up the hierarchy. For example, for job titles, the jobs "Plumber" and "Bricklayer" could be generalised to "Tradesperson". The second option is to simply replace the individual values with a range set. For example, in a region where job titles can be generalised and all tuples contain either "Plumber" or "Bricklayer", these could be replaced with "[Plumber, Bricklayer]". The resulting generalised regions can be merged to create a k-anonymisation of *D*.

Regarding investigation of the utility of the anonymised data, as mentioned, a machine learning algorithm will be used to measure how useful the post-anonymisation data is in analysis. It is likely that the machine learning algorithm will be a decision tree however the specific algorithm has not been decided on at this stage. The general idea will be to run the decision tree algorithm on multiple sets of test data and measure the accuracy of the classification. Following this, the same test data will be anonymised using the tool created in the project for different values of k, the same decision tree algorithm will then be used again on this anonymised data for classification. The accuracy of the pre- and post-anonymisation classifications will be compared and an evaluation based upon the results will be provided in the final report.

Aims & Objectives

The following are my overall aims and objectives for this proposed project for the full duration of the project timeframe:

1. Implement a k-anonymisation tool for relational data
    - Develop a deep understanding of the Mondrian k-anonymity algorithm
    - Implement the algorithm with the Python programming language, ensuring the parameters can be easily modified by the user
    - Ensure the tool is lightweight and relatively user-friendly
    - Test the tool on multiple different sets of data repeatedly

    The resulting tool will be used in the next objective.

2. Measure the usefulness of the anonymised data provided by the tool using a machine learning algorithm
    - Identify a specific machine learning algorithm that would be suitable, and gain a deep understanding of the selected algorithm
    - Apply the algorithm to pre-anonymisation data for multiple data sets and collect results
    - Apply the algorithm to post-anonymisation data for multiple data sets and collect results

3. Evaluate the results collected from Objective 2 and provide any possible insight into the usefulness of anonymised data regarding analysis
    - Collate collected data, providing relevant figures and graphs in report
    - Examine the data and compare the effectiveness of anonymisation relative to the usefulness of the data in analysis
    - Provide a report of the evaluation and any insights found

The result of the project will be the tool created in Objective 1 and a final report containing details on my implementation, results of measurements from Objective 2, and any findings from the evaluation from Objective 3.

A meeting has been scheduled with the supervisor once per week to discuss the project progress. It is possible that the details and the approach in this report are subject to change based on discussion from these meetings. Any changes will be highlighted in the final report.

The initial work plan timeline for the project is on the following page.

Implementation of a Data Privacy Protection Tool for Relational Data

Implementation & Evaluation Timeline

2021

	Feb				March				April					May	
	W1	W2	W3	W4	W5	W6	W7	W8	E1	E2	E3	W9	W10	W11	W12
Write initial plan	5th Feb														
Background research & understanding		12th Feb													
Implement Mondrian k-anonymity algorithm in Python				26th Feb											
Integrate k-anonymity algorithm into user friendly tool					2nd March										
Testing of k-anonymity tool					5th March										
Milestone 1 - k-anonymity tool complete					5th March										
Implement machine learning algorithm							17th March								
Apply machine learning algorithm to pre/post-anonymisation data & measure effectiveness								26th March							
Free buffer time for catch-up															
Milestone 2 - measurement of machine learning effectiveness on anonymised data complete											18th April				
Collate data collected and examine results													26th April		
Write final report															14th May
Milestone 3 - evaluation of results and final report complete															14th May

References

[1] Friedman, J H. et al. 1977. *An Algorithm for Finding Best Matches in Logarithmic Expected Time*. ACM Transactions on Mathematical Software, Vol. 3(3), pp. 209-226.

[2] Fung, B C M. et al. 2010. *Privacy-preserving data publishing: A survey of recent developments.* ACM Computing Surveys.

[3] LeFevre, K. et al. 2006. *Mondrian Multidimensional K-Anonymity*. University of Wisconsin, Madison.

[4] Samarati, P. and Sweeney, L. 1998. *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression*. SRI International.